

Εκτίμηση Βάθους (Depth Estimation), stereo vision και 3D Ανακατασκευή (3D Reconstruction)



Μια εισαγωγή στη Γεωμετρική Υπολογιστική Όραση (Geometric Computer Vision)

Γ. Στεφανής Τεχνητή Όραση (9.010)

- 🎬 Κίνητρο και εφαρμογές: γιατί χρειαζόμαστε βάθος (depth);
- 🎬 Μοντέλο κάμερας (pinhole) & βαθμονόμηση (calibration)
- 🎬 Στερεοσκοπική γεωμετρία (epipolar geometry) & ορθοαναγωγή (rectification)
- 🎬 Stereo matching → ανομοιότητα (disparity) → βάθος (depth)
- 🎬 3D ανακατασκευή: τριγωνισμός (triangulation) και νέφος σημείων (point cloud)
- 🎬 Σύνδεση με Φωτογραμμετρία (photogrammetry) / SfM και σύγχρονες προσεγγίσεις
- 🎬 Datasets/αξιολόγηση και πρακτικές παγίδες

Κίνητρο: γιατί «βάθος»;

Applications που το κάνουν απαραίτητο

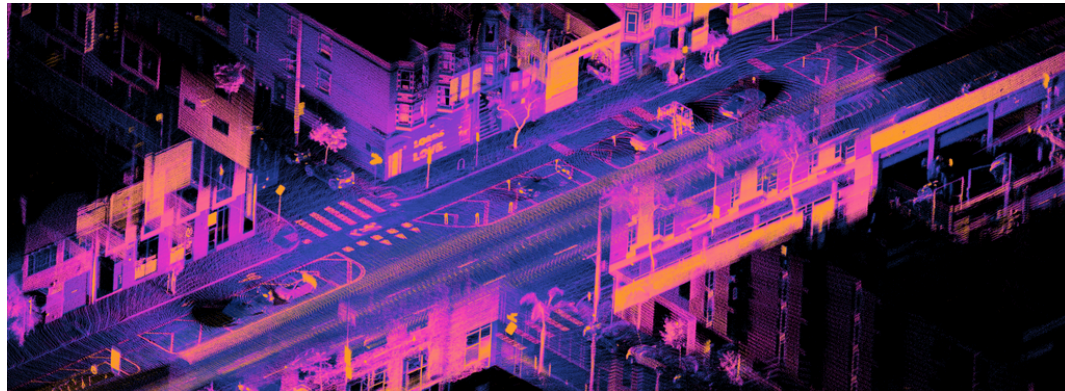
Το βάθος (depth) μετατρέπει την εικόνα από 2D παρατήρηση σε 3D κατανόηση.



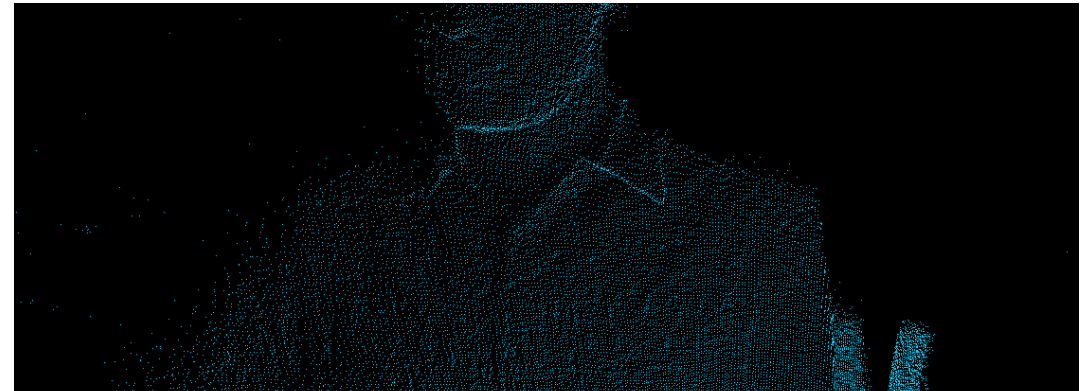
UAV χαρτογράφηση (UAV mapping)



Βιομηχανική όραση (industrial vision)



Αυτόνομη οδήγηση (autonomous driving)



Depth sensors (ToF) / AR

Τι είναι χάρτης βάθους;

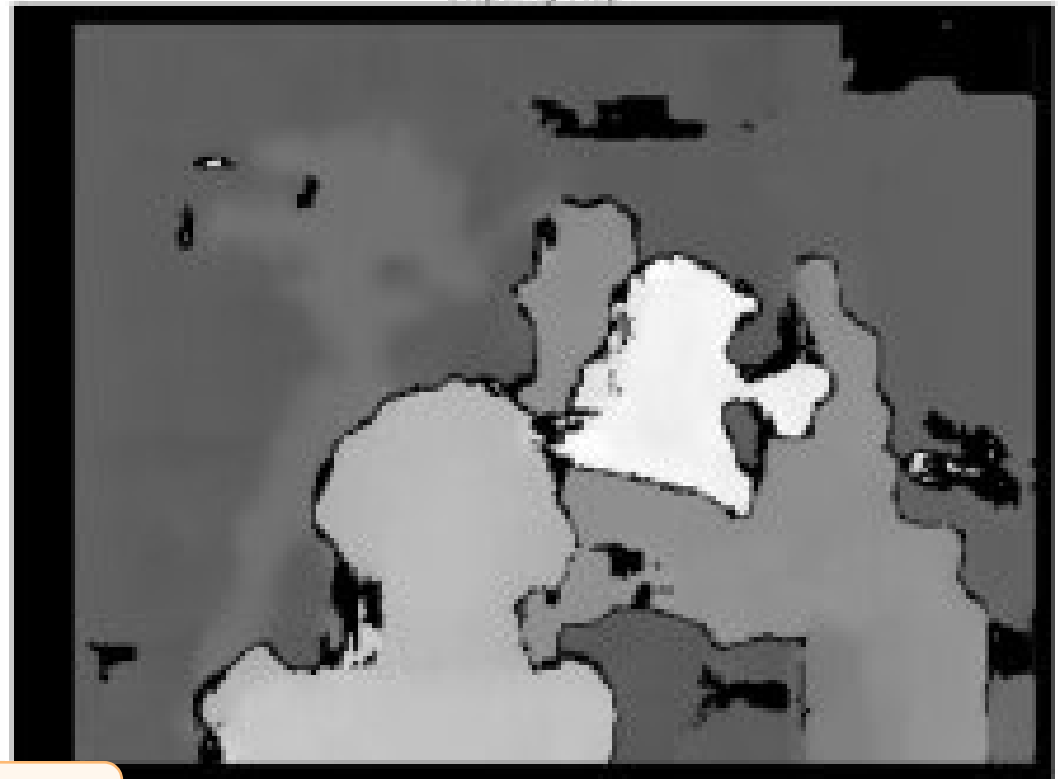
Depth map / disparity map σε 1 εικόνα

Χάρτης βάθους (depth map): για κάθε pixel δίνει απόσταση (Z) ή σχετικό βάθος.
Στο στερεοζεύγος, πρώτα εκτιμούμε ανομοιότητα (disparity) και μετά μετατρέπουμε σε βάθος.

Original Image



Disparity Map



Αριστερά: αρχική εικόνα • Δεξιά: χάρτης ανομοιότητας (disparity map)

Τρόποι εκτίμησης βάθους (Depth Estimation)

Passive / Active / Learning-based

Παθητικές μέθοδοι (Passive): βασίζονται σε εικόνες

Stereo (2 κάμερες)

Passive

- 🎬 Dense disparity → depth
- 🎬 Απαιτεί calibration & rectification
- 🎬 Δύσκολο σε textureless/reflective

Multi-view / SfM + MVS

Passive

- 🎬 Πολλές εικόνες → 3D (sparse/dense)
- 🎬 Robust estimation (RANSAC) + BA
- 🎬 Scale ambiguity χωρίς πρόσθετα cues

Ενεργές & learning-based μέθοδοι

LiDAR / ToF / Structured light

Active

- 🎬 Άμεσες μετρήσεις range
- 🎬 Αραιά/θορυβώδη → depth completion
- 🎬 Κόστος/ενέργεια/συνθήκες φωτισμού

Learning-based (CNN/ViT)

Learning

- 🎬 Monocular depth από 1 εικόνα
- 🎬 Supervised ή self-supervised training
- 🎬 Κλίμακα/γενίκευση είναι συχνά το bottleneck

Φωτογραμμετρία (photogrammetry): τεχνικές μετρήσεων και ανακατασκευής (2D/3D) από φωτογραφίες με στόχο γεωμετρική ακρίβεια (π.χ. χαρτογράφηση, τοπογραφία, πολιτιστική κληρονομιά).

- 🎬 Structure-from-Motion (SfM): photogrammetry + computer vision για ταυτόχρονη εκτίμηση πόζας καμερών (camera poses) και 3D δομής (sparse point cloud).
- 🎬 Multi-View Stereo (MVS): «πυκνώνει» το αποτέλεσμα σε dense depth / dense reconstruction.
- 🎬 Για βαθμονομημένο stereo, η κλίμακα είναι γνωστή (baseline B) → μετρήσεις σε μέτρα.

Stereo Vision: Πλήρες Pipeline 3D Ανακατασκευής

Calibration → Rectification → Matching → Disparity → Depth → Point Cloud

1) Calibration
(K, distortion,
baseline B)

▶ 2) F/E/H
+ Rectification

▶ 3) Stereo
Matching cost
(C)

▶ 4) Regularization
SGM / SGBM

▶ 5) Disparity
map $d(x,y)$

▶ 6) Depth
 $Z = fB / d$

▶ 7) Point Cloud
/ Mesh

Κύριες έξοδοι:

- Rectified stereo pair
- Disparity map (πυκνό)
- Metric depth map
- 3D point cloud / surface

Σημ.: Για metric βάθος χρειάζεται σωστό baseline B και intrinsics K.
Για 2-view geometry: F (uncalibrated), E (calibrated), H (planar/rotation).

Τι κάνει κάθε στάδιο

Inputs/outputs + πρακτικές παγίδες (failure modes)

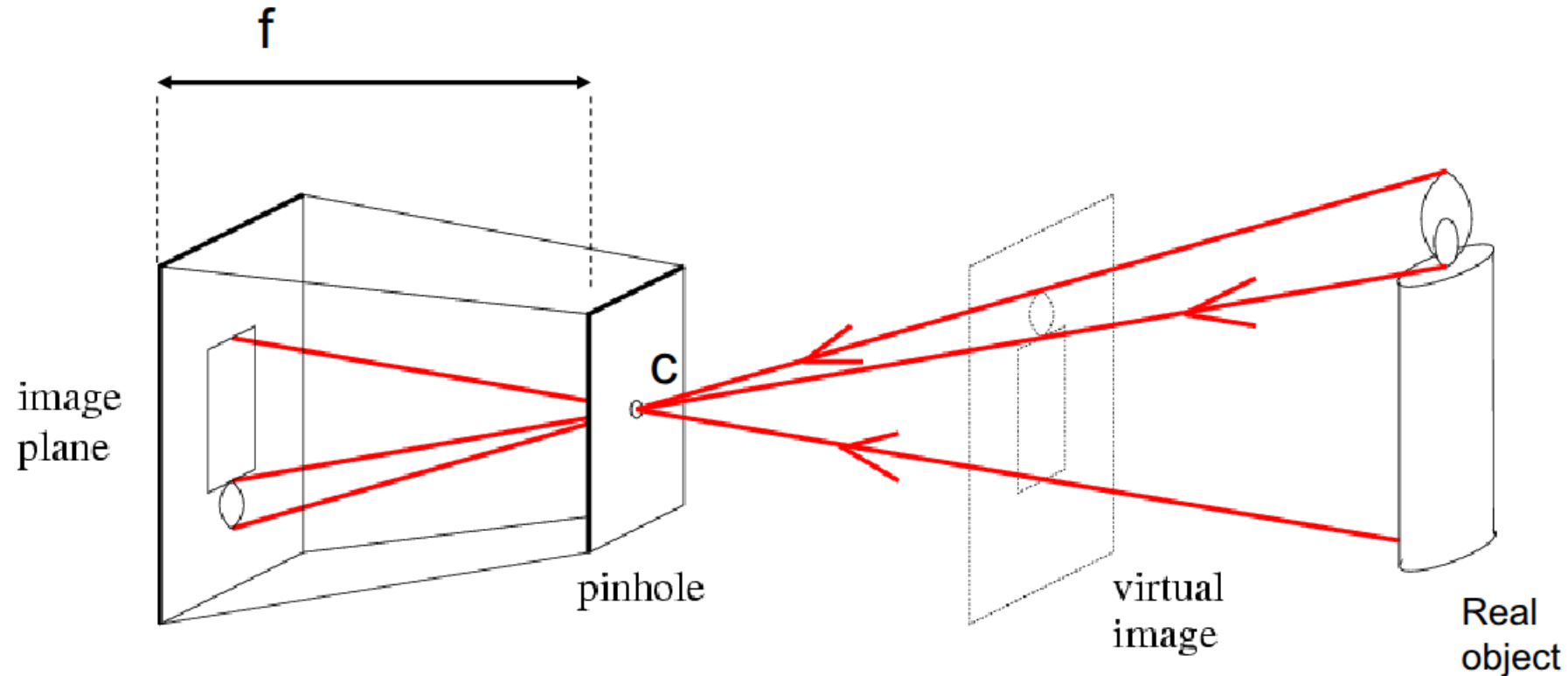
Στάδια (inputs → outputs)

- Calibration: checkerboard/rig → K , distortion, baseline B
- Undistort: raw images → διορθωμένες εικόνες
- Rectification: $F/E (+K, R, t)$ → οι επιτολικές ευθείες γίνονται οριζόντιες
- Matching cost: (L, R) → $C(x, y, d)$ (π.χ. SAD, Census, NCC)
- Regularization (SGM): C → συνεπές disparity (τιμωρία ασυνέχειας)
- Disparity → Depth: $Z = fB/d$ (ή $Z = fB/(d+d_{offs})$ σε datasets)
- Depth → 3D: back-projection → point cloud / mesh

Κρίσιμες παγίδες

- Λάθος calibration → λάθος depth κλίμακα (συστηματικό σφάλμα)
- Occlusions → “τρύπες” στο disparity (L-R consistency check)
- Textureless περιοχές → ασαφές minimum στο $C(d)$
- Repetitive patterns → λάθος αντιστοιχίσεις (ambiguity)
- Specular/lighting changes → mismatch (χρήσιμα: Census/robust costs)
- Sub-pixel disparity → καλύτερο Z (αλλά ευαίσθητο σε θόρυβο)
- Depth error grows with distance: $Z \sim 1/d$ (μακριά χειροτερεύει)

Pinhole camera model



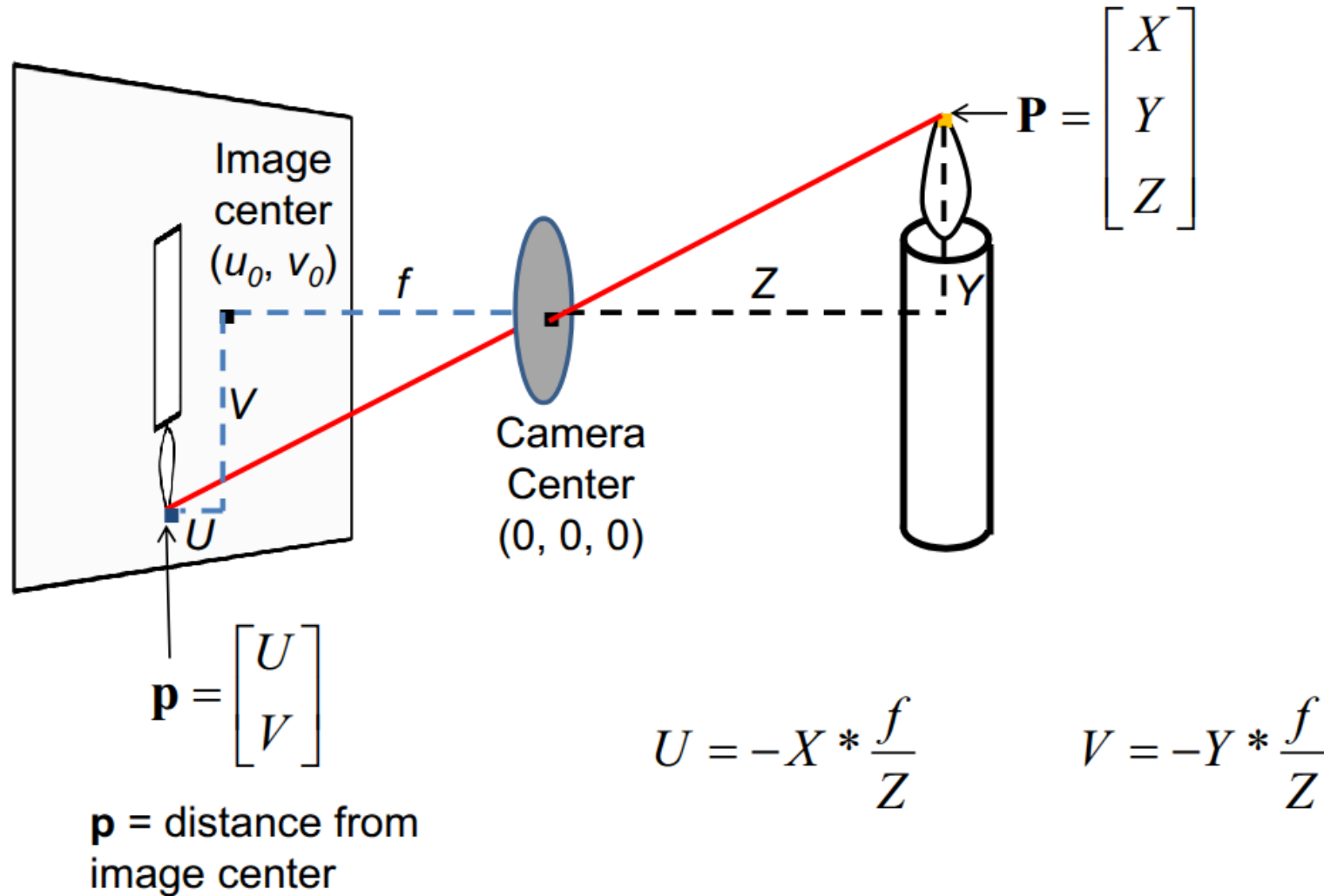
f = Focal length

c = Optical center of the camera

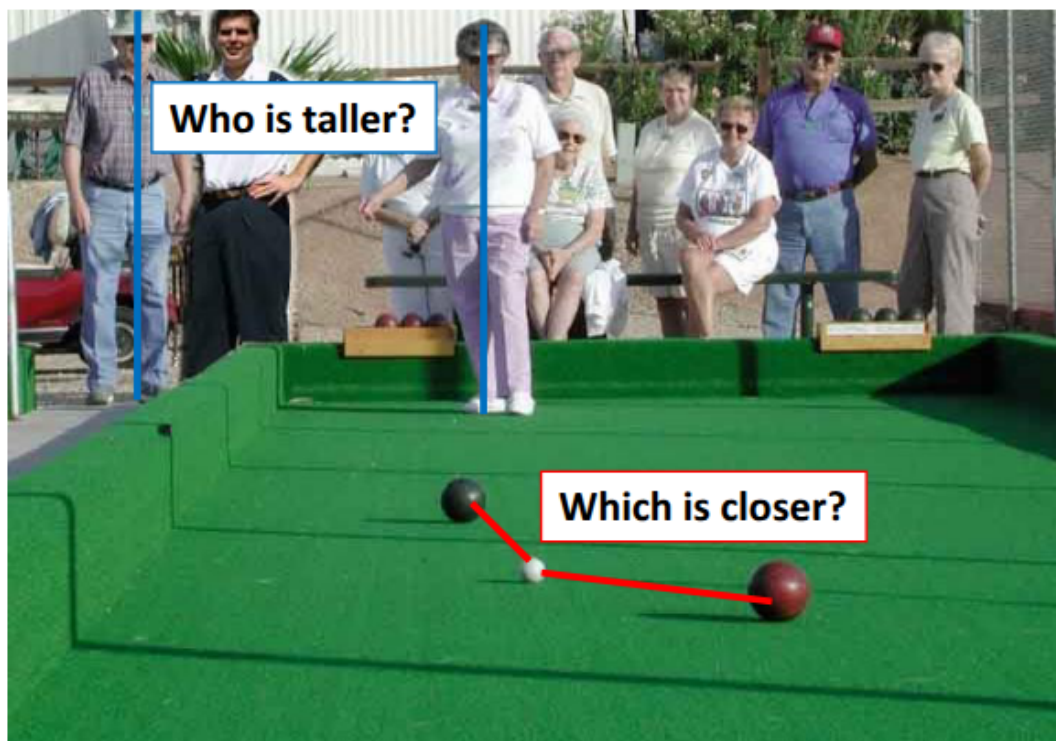
Μοντέλο κάμερας (Pinhole model)

Intrinsics (K) / Extrinsics (R,t)

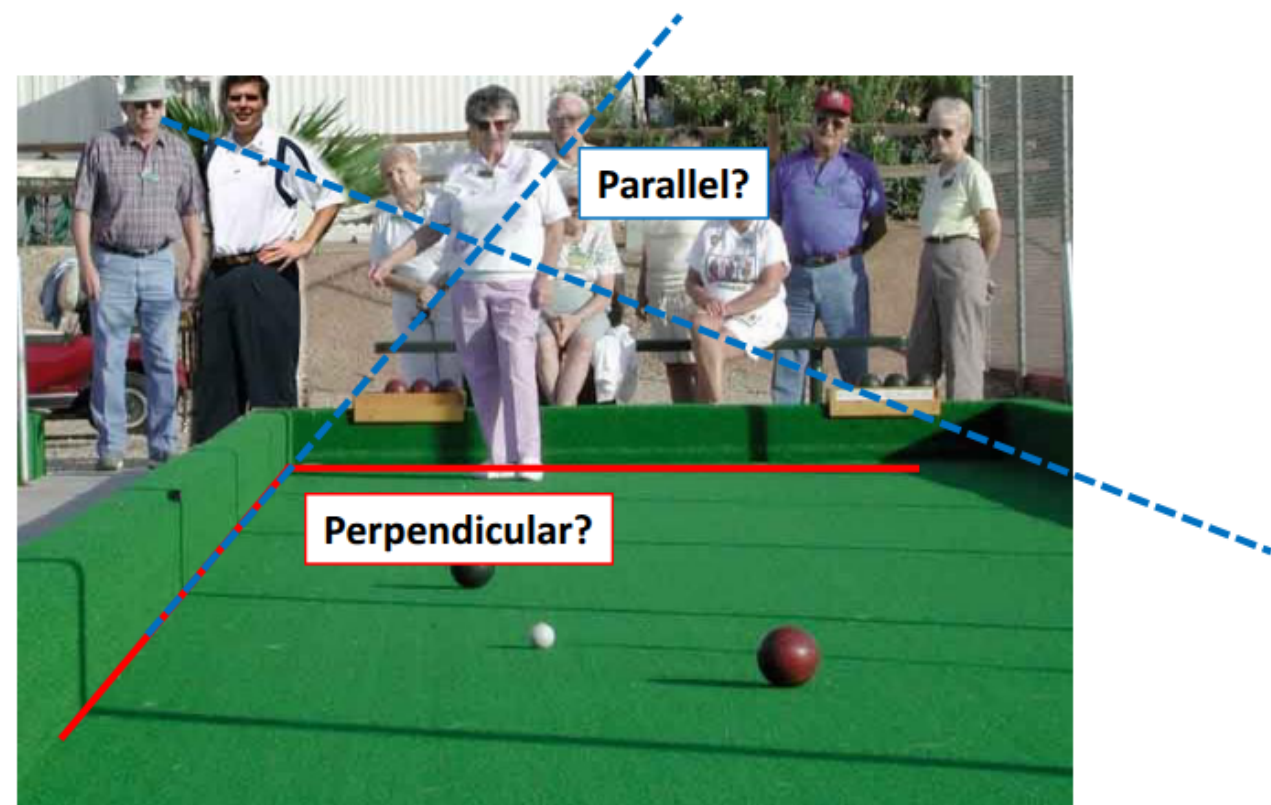
Projection: world coordinates \rightarrow image coordinates



Length (and so area) is lost.

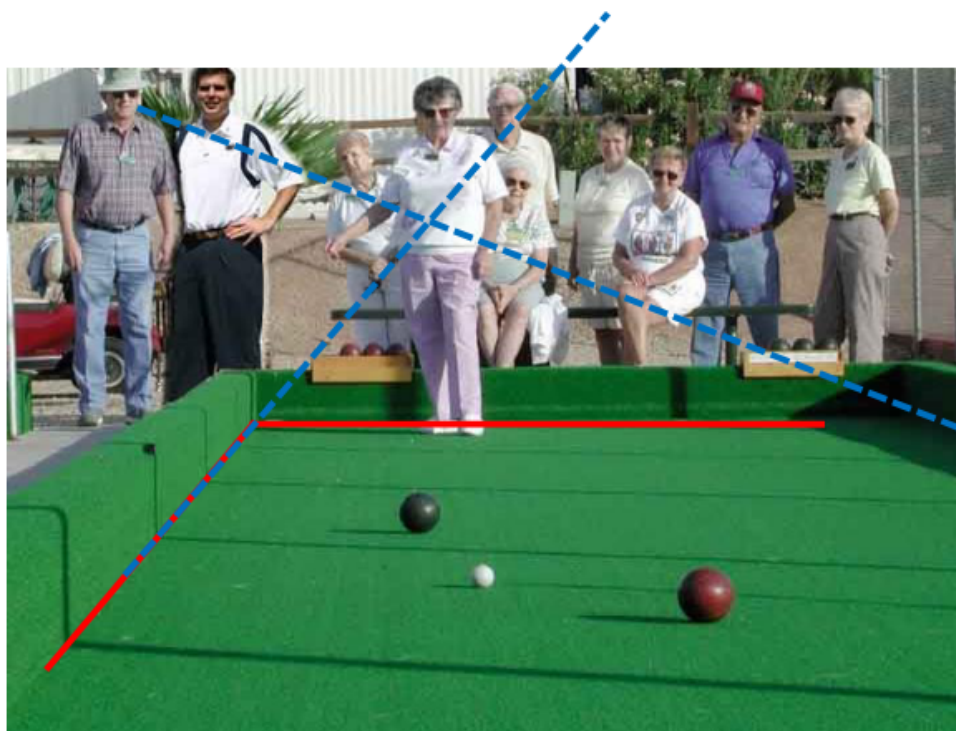


Angles are lost.

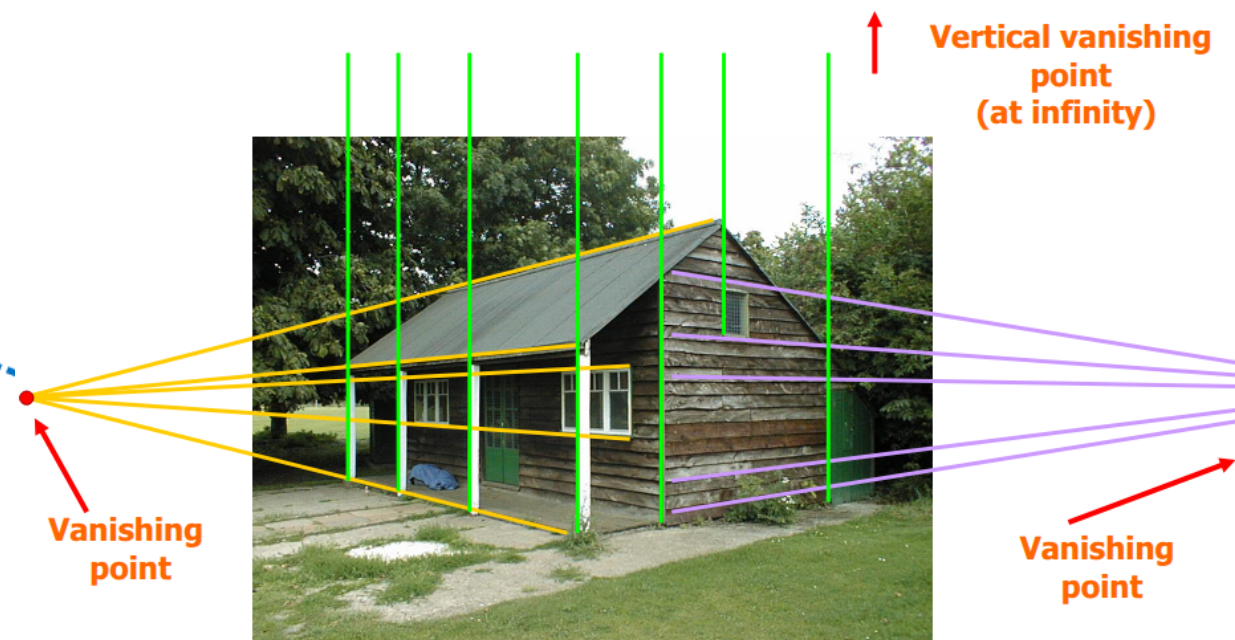


What is preserved?

- Straight lines are still straight.



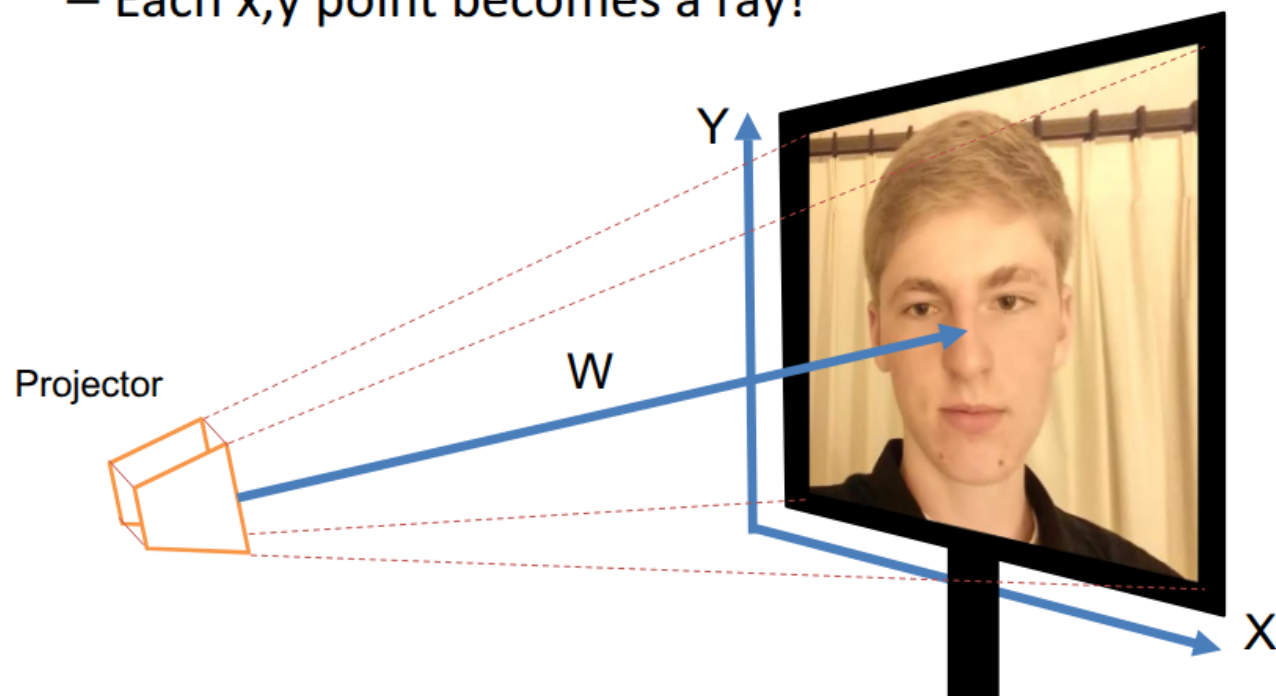
Vanishing points and lines



Slide from Efros, Photo from Criminisi

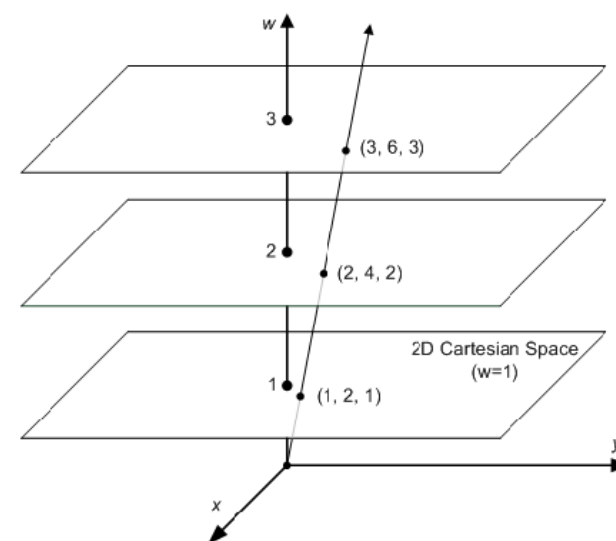
Projective geometry

- 2D point in projective = (x,y,w) coordinates
 - w defines the scale of the projected image.
 - Each x,y point becomes a ray!



Homogeneous coordinates

- Projective
- Point becomes a line



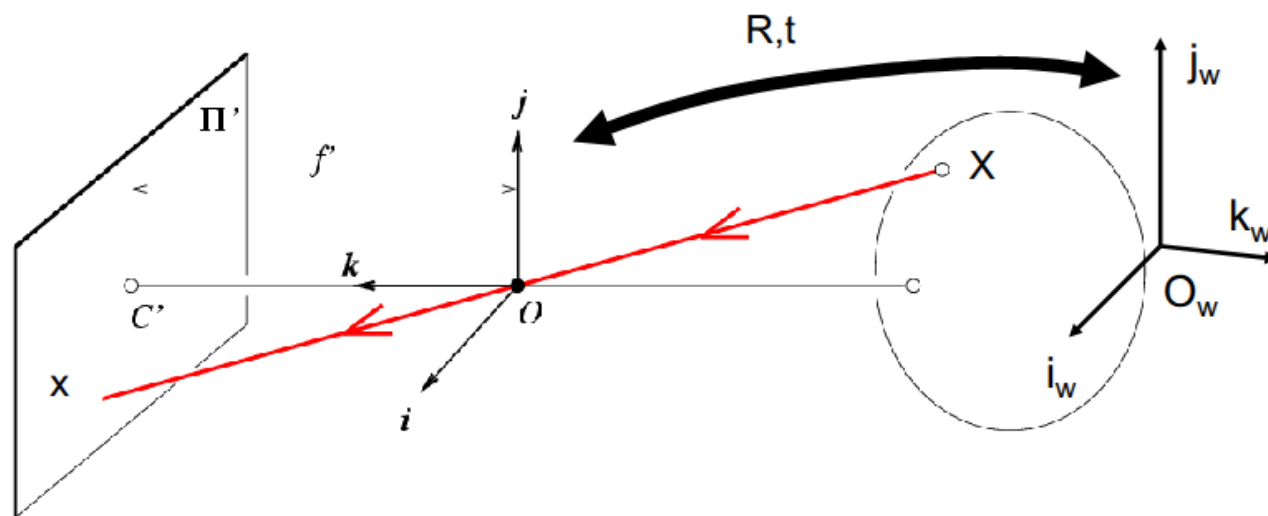
To homogeneous

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

From homogeneous

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

Camera (projection) matrix



$$\mathbf{x} = \mathbf{K} \underbrace{\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}}_{\text{Extrinsic Matrix}} \mathbf{X}$$

Extrinsic Matrix

\mathbf{x} : Image Coordinates: $(u, v, 1)$

\mathbf{K} : Intrinsic Matrix (3×3)

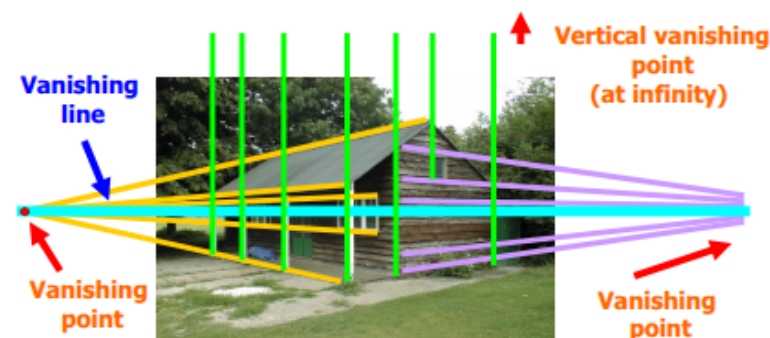
\mathbf{R} : Rotation (3×3)

\mathbf{t} : Translation (3×1)

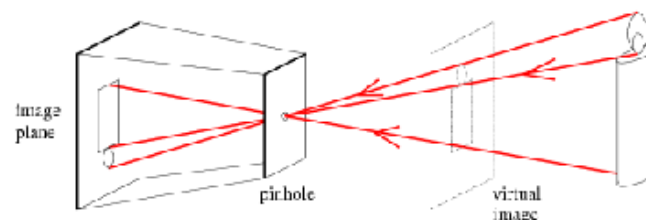
\mathbf{X} : World Coordinates: $(X, Y, Z, 1)$

Projective geometry

Projection loses length, area, angle, but straight lines remain straight.



Pinhole camera model and camera projection matrix.



$$\mathbf{x} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \mathbf{X}$$

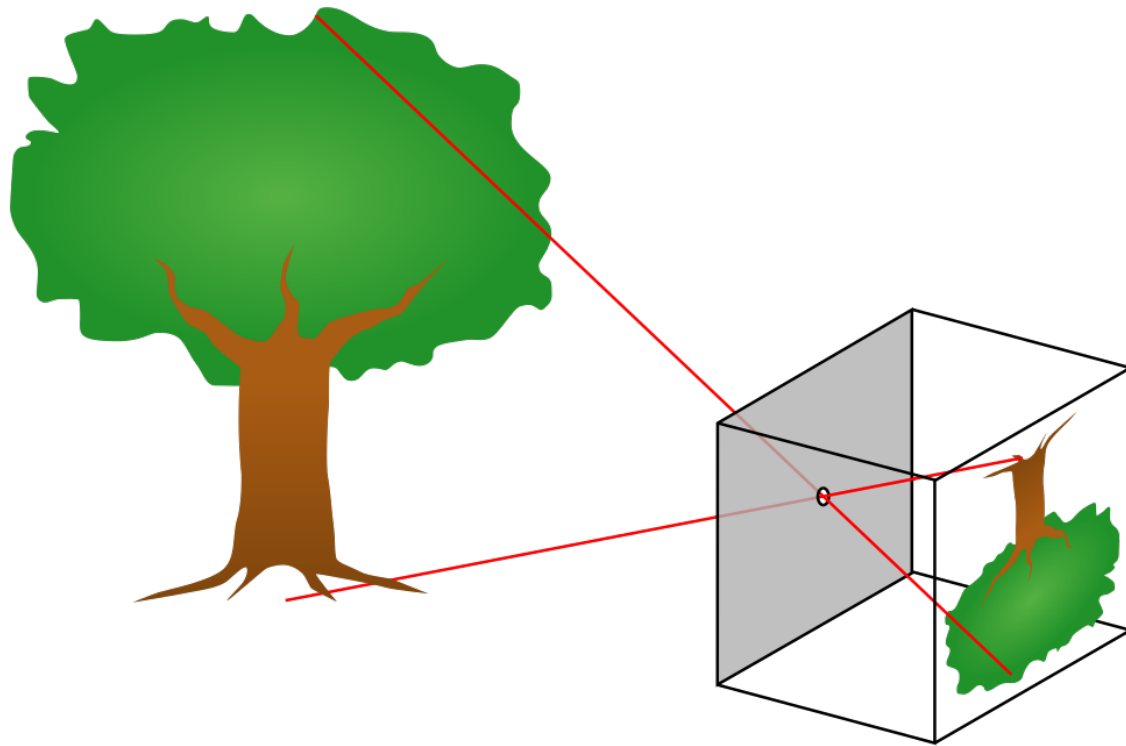
Homogeneous coordinates.

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

James Hays

Μοντέλο κάμερας (Pinhole model)

Intrinsics (K) / Extrinsics (R,t)



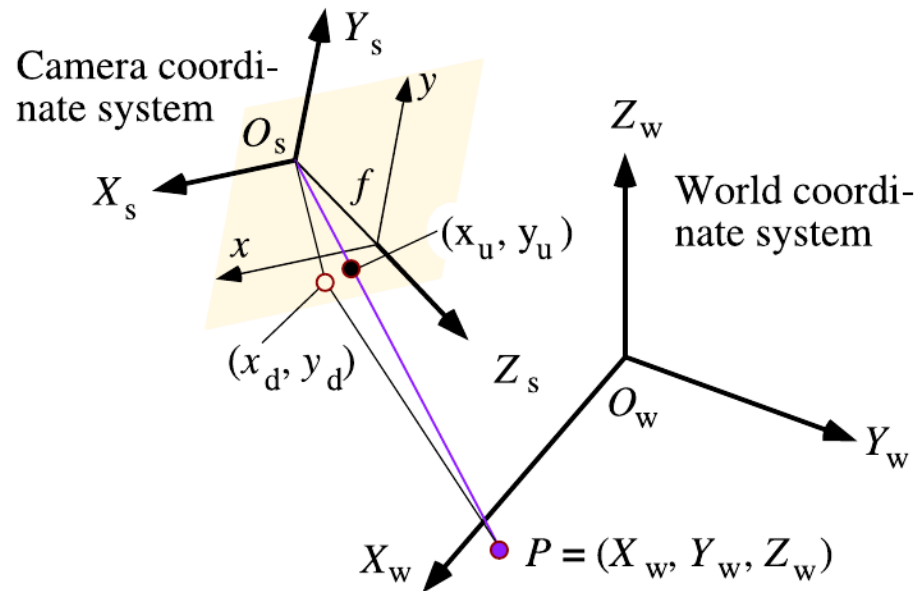
Τι κρατάμε:

- 🎬 Προβολή προοπτικής (perspective projection): $3D \rightarrow 2D$.
- 🎬 Εσωτερικές παράμετροι (intrinsics): f , (c_x, c_y) , skew, κλίμακα pixel.
- 🎬 Εξωτερικές (extrinsics): περιστροφή R και μετατόπιση t (πόζα κάμερας).
- 🎬 Παραμορφώσεις φακού (lens distortion): πρέπει να διορθωθούν πριν από stereo matching.

Στην εκτίμηση βάθους, η βαθμονόμηση (calibration) είναι τόσο σημαντική όσο και το matching.

Μοντέλο κάμερας (Pinhole model)

Intrinsics (K) / Extrinsics (R,t)



Τι κρατάμε:

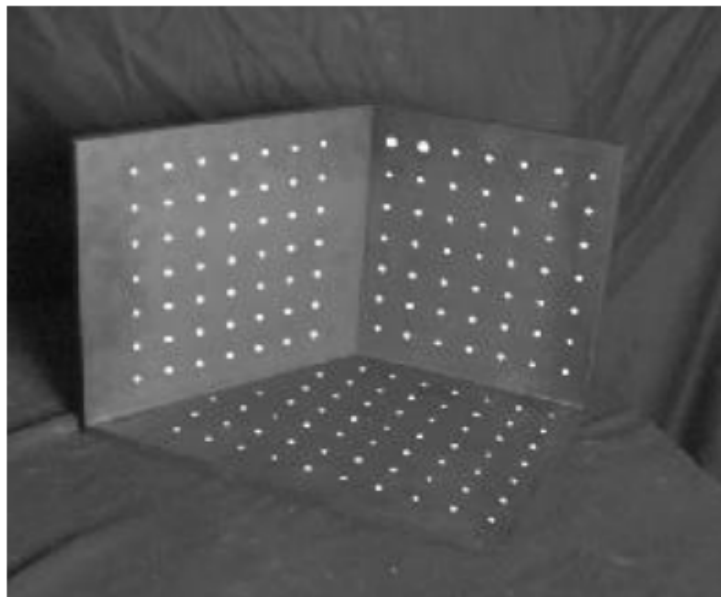
- 🎬 Προβολή προοπτικής (perspective projection): 3D → 2D.
- 🎬 Εσωτερικές παράμετροι (intrinsics): f, (c_x, c_y), skew, κλίμακα pixel.
- 🎬 Εξωτερικές (extrinsics): περιστροφή R και μετατόπιση t (πόζα κάμερας).
- 🎬 Παραμορφώσεις φακού (lens distortion): πρέπει να διορθωθούν πριν από stereo matching.

Στην εκτίμηση βάθους, η γεωμετρία (calibration) είναι τόσο σημαντική όσο και το matching.

Calibrating the Camera

Use an scene with **known** geometry

- Correspond image points to 3d points
- Get least squares solution (or non-linear solution)



Known 2d
image coords

Known 3d
world locations

$$\begin{matrix} \downarrow & & & & \downarrow \\ \begin{bmatrix} su \\ sv \\ s \end{bmatrix} & = & \begin{matrix} \mathbf{M} \\ \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \end{matrix} & \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \\ \uparrow & & & & \end{matrix}$$

Unknown Camera Parameters

Unknown Camera Parameters

↓

$$\begin{matrix} \text{Known 2d} \\ \text{image coords} \end{matrix} \begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \begin{matrix} \text{Known 3d} \\ \text{locations} \end{matrix}$$

Next, rearrange into form where all **M** coefficients are individually stated in terms of X,Y,Z,u,v.

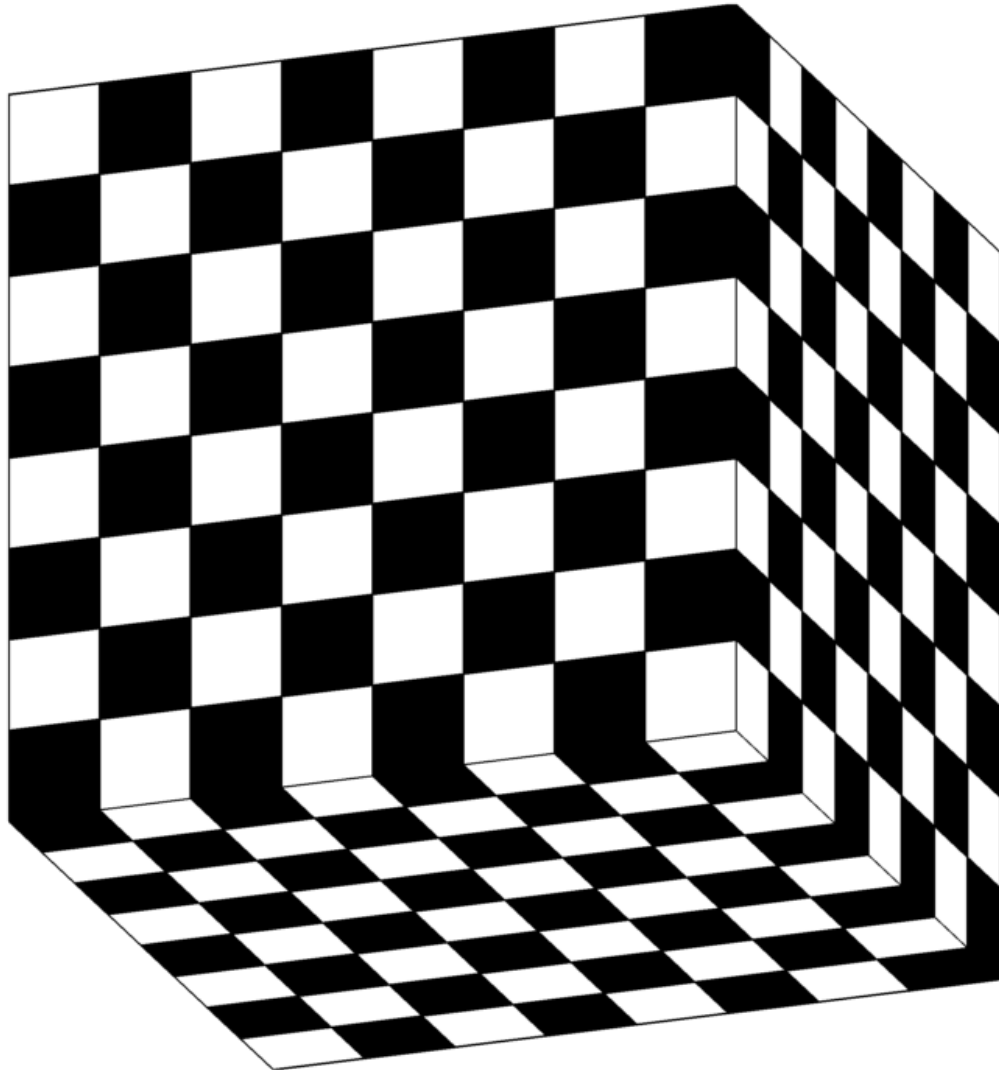
-> Allows us to form lsq matrix.

$$m_{31}uX + m_{32}uY + m_{33}uZ + m_{34}u = m_{11}X + m_{12}Y + m_{13}Z + m_{14}$$

$$m_{31}vX + m_{32}vY + m_{33}vZ + m_{34}v = m_{21}X + m_{22}Y + m_{23}Z + m_{24}$$

$$0 = m_{11}X + m_{12}Y + m_{13}Z + m_{14} - m_{31}uX - m_{32}uY - m_{33}uZ - m_{34}u$$

$$0 = m_{21}X + m_{22}Y + m_{23}Z + m_{24} - m_{31}vX - m_{32}vY - m_{33}vZ - m_{34}v$$



Τι εκτιμούμε (estimate):

- 🎬 K (intrinsics): $f_x, f_y, (c_x, c_y)$
- 🎬 Παραμόρφωση φακού (radial/tangential distortion)
- 🎬 Stereo extrinsics: R, t μεταξύ αριστερής/δεξιάς κάμερας
- 🎬 Baseline B (γνωστό) → απόλυτη κλίμακα βάθους

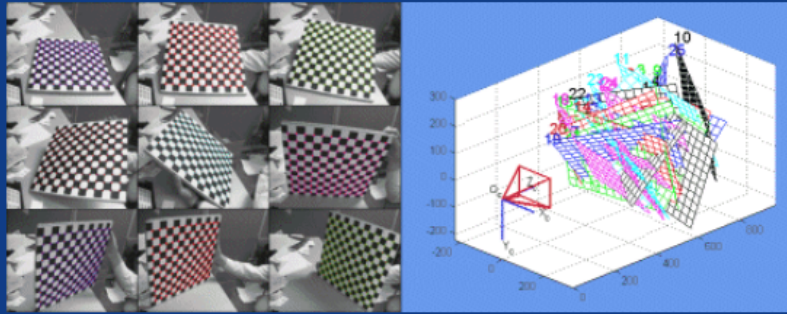
Πρακτικά:

- 🎬 Πολλές λήψεις/πόζες chessboard → σταθερότερη εκτίμηση (robust).
- 🎬 Undistort + Rectify πριν από disparity.

Βαθμονόμηση (Calibration)

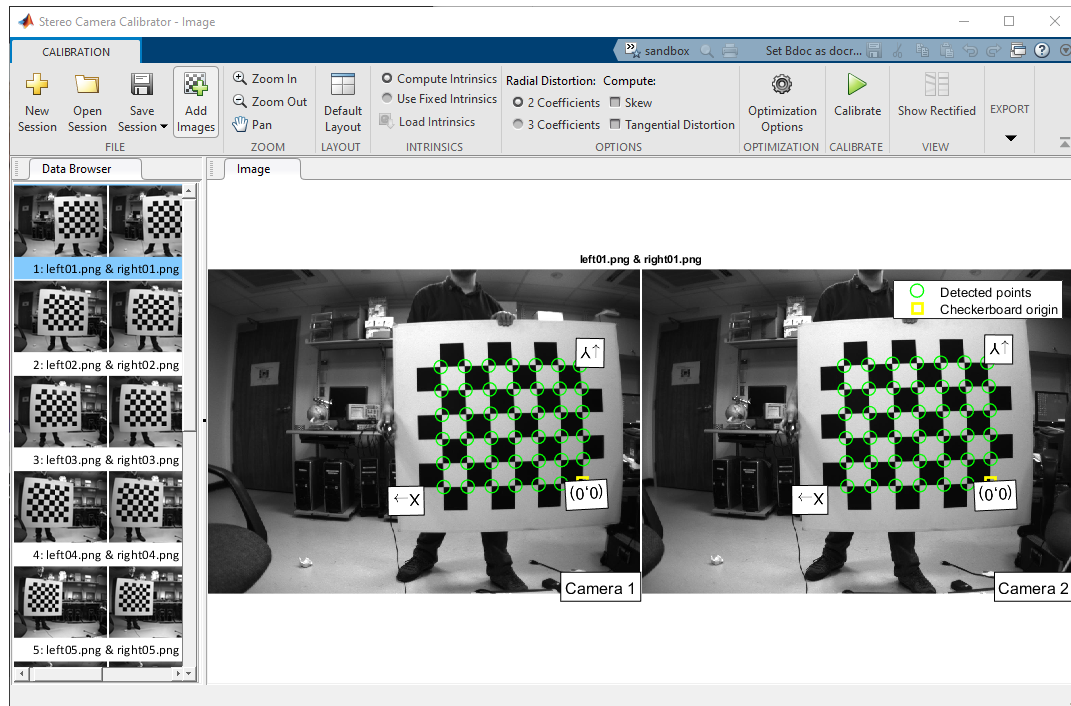
Intrinsics, distortion, stereo extrinsics

Camera Calibration Toolbox for Matlab



Τι εκτιμούμε (estimate):

- K (intrinsics): $f_x, f_y, (c_x, c_y)$
- Παραμόρφωση φακού (radial/tangential distortion)
- Stereo extrinsics: R, t μεταξύ αριστερής/δεξιάς κάμερας
- Baseline B (γνωστό) → απόλυτη κλίμακα βάθους

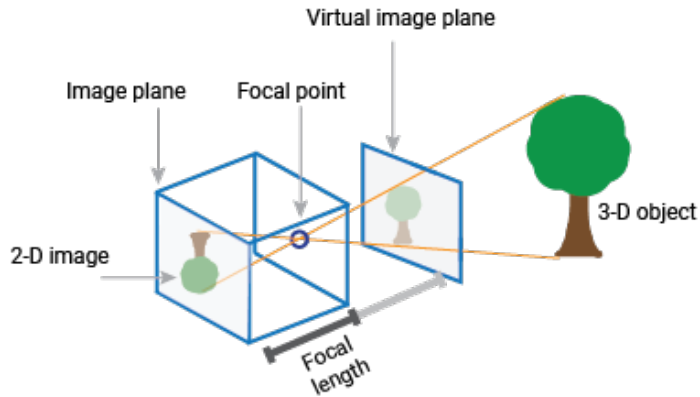


Πρακτικά:

- Πολλές λήψεις/πόζες chessboard → σταθερότερη εκτίμηση (robust).
- Undistort + Rectify πριν από disparity.

Βαθμονόμηση (Calibration)

Intrinsics, distortion, stereo extrinsics

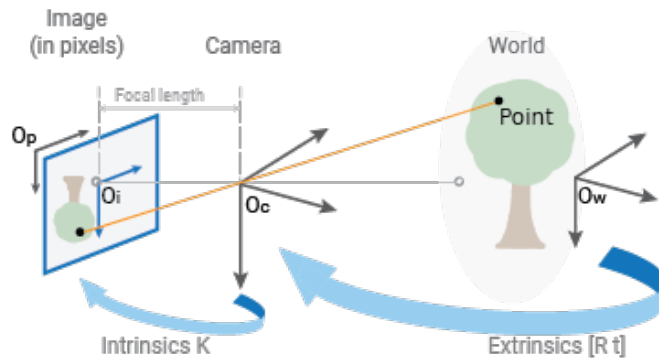


$$W \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Scale factor W Image points $\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$ World points $\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$

$$P = K \begin{bmatrix} R & t \end{bmatrix}$$

Camera matrix P Intrinsic matrix K Extrinsic Rotation and Translation $\begin{bmatrix} R & t \end{bmatrix}$



Τι εκτιμούμε (estimate):

- 🎬 K (intrinsics): $f_x, f_y, (c_x, c_y)$
- 🎬 Παραμόρφωση φακού (radial/tangential distortion)
- 🎬 Stereo extrinsics: R, t μεταξύ αριστερής/δεξιάς κάμερας
- 🎬 Baseline B (γνωστό) \rightarrow απόλυτη κλίμακα βάθους

Πρακτικά:

- 🎬 Πολλές λήψεις/πόζες chessboard \rightarrow σταθερότερη εκτίμηση (robust).
- 🎬 Undistort + Rectify πριν από disparity.

Βαθμονόμηση (Calibration)

Intrinsics, distortion, stereo extrinsics

Τι εκτιμούμε (estimate):

Examples of what you can do after calibrating your camera:

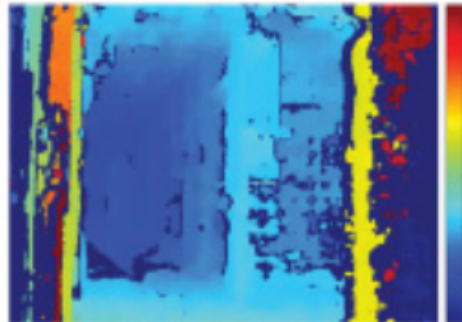
Before



After



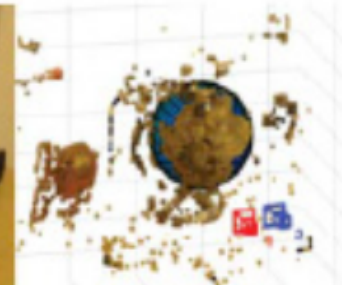
Remove Lens Distortion



Estimate Depth Using
a Stereo Camera



Measure Planar
Objects



Estimate 3-D Structure from
Camera Motion

Essential Matrix και Fundamental Matrix

Στην υπολογιστική όραση, ο Essential Matrix (Ουσιώδης Πίνακας) και ο Fundamental Matrix (Θεμελιώδης Πίνακας) είναι δύο εργαλεία-κλειδιά για την περιγραφή της γεωμετρίας μεταξύ δύο εικόνων (Epipolar Geometry).

Παρόλο που και οι δύο πίνακες συνδέουν αντίστοιχα σημεία από δύο διαφορετικές λήψεις της ίδιας σκηνής, η κύρια διαφορά τους έγκειται στο αν γνωρίζουμε τις εσωτερικές παραμέτρους της κάμερας.

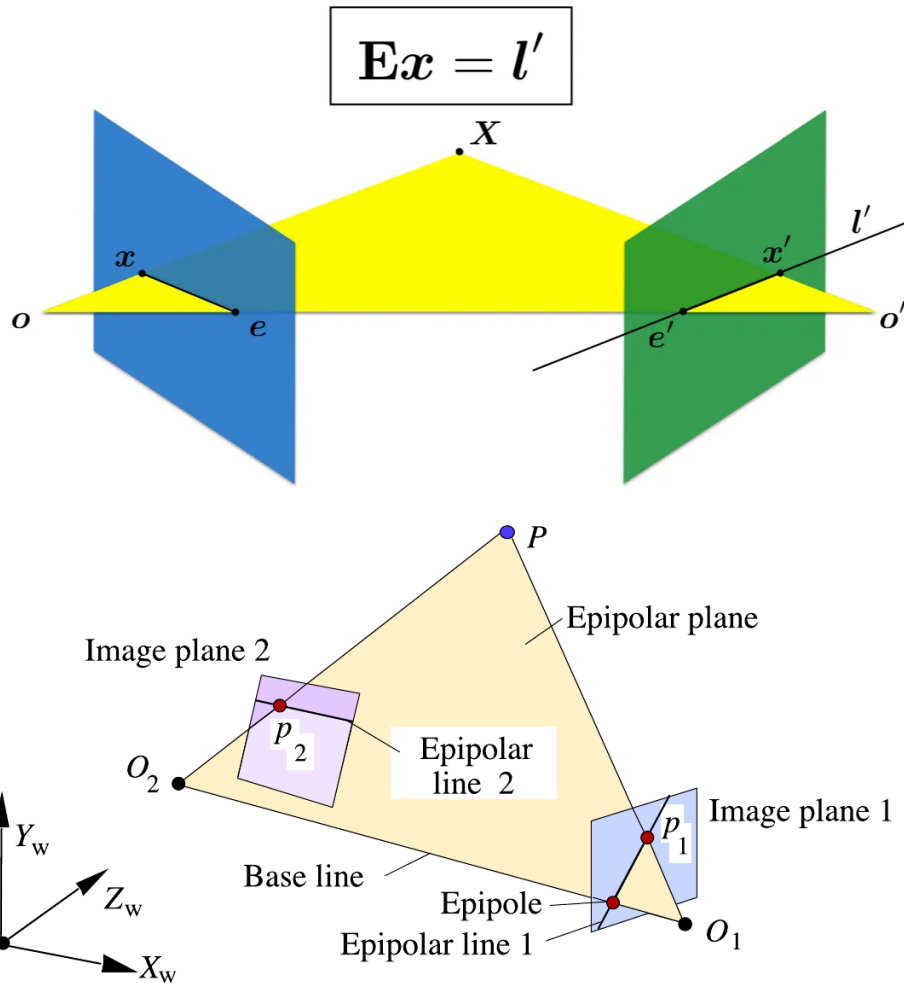
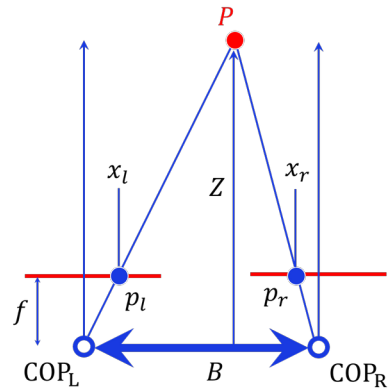
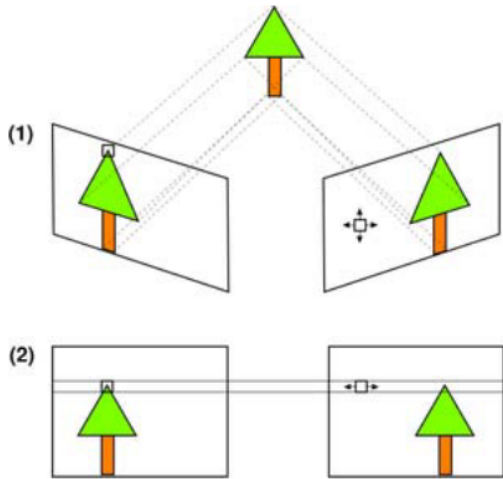


Fig. 7.15 A sketch of epipolar geometry for two cameras in general poses



$$\begin{bmatrix} p' \\ x' & y' & f \end{bmatrix} \begin{bmatrix} E \\ 0 & 0 & 0 \\ 0 & 0 & B \\ 0 & -B & 0 \end{bmatrix} \begin{bmatrix} p \\ x \\ y \\ f \end{bmatrix}$$

$$Bfy' = Bfy \Rightarrow y' = y$$

Given a known point (x,y) in the original image, this is a line in the (x',y') image

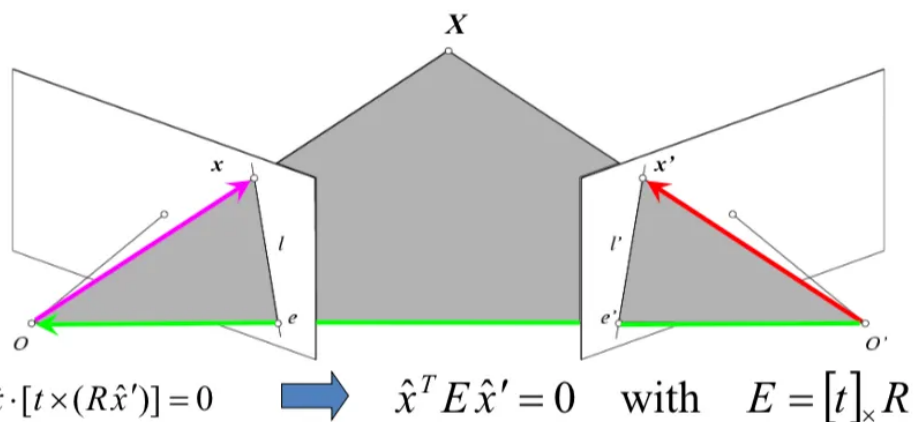
Essential Matrix και Fundamental Matrix

Η πρακτική σημασία των δύο αυτών πινάκων στο Stereo Vision και στο Depth Estimation (εκτίμηση βάθους) είναι καθοριστική, καθώς αποτελούν το μαθηματικό θεμέλιο για να μετατρέψουμε δύο επίπεδες εικόνες (2D) σε τρισδιάστατη πληροφορία (3D).

Θεμελιώδης - Fundamental Matrix (F): Είναι ο "χάρτης" που μας επιτρέπει να αντιστοιχίζουμε pixels μεταξύ των δύο καμερών χωρίς να ξέρουμε τίποτα γι' αυτές. Είναι το πρώτο βήμα για να βρούμε την παράλλαξη (disparity).

Ουσιώδης - Essential Matrix (E): Είναι το εργαλείο που μετατρέπει τις αντιστοιχίες των pixels σε πραγματική γεωμετρική πληροφορία (μέτρα, γωνίες), επιτρέποντας την ακριβή μέτρηση αποστάσεων στο 3D χώρο.

Με απλά λόγια: Χωρίς αυτούς τους πίνακες, η κάμερα θα έβλεπε απλά δύο άσχετες εικόνες. Με αυτούς, "καταλαβαίνει" τη γεωμετρική σχέση τους και μπορεί να υπολογίσει ότι ένα αντικείμενο βρίσκεται π.χ. ακριβώς στα 5 μέτρα.



$$\mathbf{a} = (a_1 \ a_2 \ a_3)^T$$

$$\mathbf{b} = (b_1 \ b_2 \ b_3)^T$$

$$[\mathbf{a}]_{\times} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \quad \mathbf{a} \times \mathbf{b} = [\mathbf{a}]_{\times} \mathbf{b}$$

[Matrix representation of the cross product](#)

↓

Essential Matrix
(Longuet-Higgins, 1981)

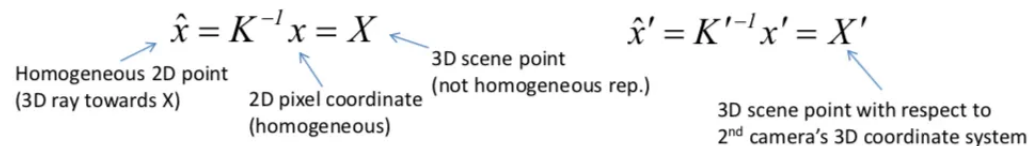
Essential Matrix

Χρησιμοποιείται όταν οι κάμερες είναι βαθμονομημένες (calibrated). Αυτό σημαίνει ότι γνωρίζουμε τον πίνακα εσωτερικών παραμέτρων K (εστιακή απόσταση, κύριο σημείο κ.λπ.)

Συντεταγμένες: Λειτουργεί σε κανονικοποιημένες συντεταγμένες εικόνας (normalized image coordinates)

Πληροφορία: Κωδικοποιεί μόνο τη σχετική πόζα (relative pose) μεταξύ των δύο καμερών, δηλαδή την περιστροφή (R) και τη μετάθεση (t).

Βαθμοί Ελευθερίας: Έχει 5 βαθμούς ελευθερίας (3 για την περιστροφή και 2 για τη μετάθεση, καθώς η κλίμακα/scale παραμένει άγνωστη)



$$\hat{x}^T E \hat{x}' = 0$$

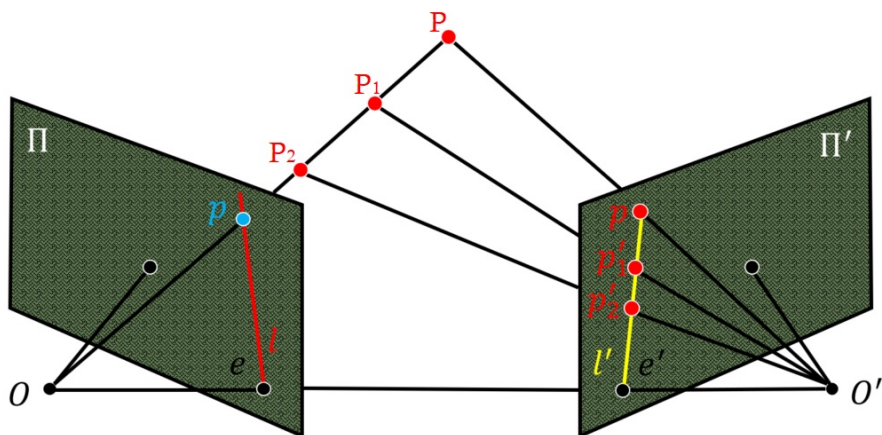
$$\hat{x} = K^{-1}x$$

$$\hat{x}' = K'^{-1}x'$$

$$\Rightarrow x^T F x' = 0 \quad \text{with} \quad F = K^{-T} E K'^{-1}$$

Fundamental Matrix
 (Faugeras and Luong, 1992)

Properties of the Fundamental Matrix



Fundamental Matrix (F)

Ο Fundamental Matrix είναι η γενίκευση του Essential Matrix για μη βαθμονομημένες κάμερες (uncalibrated).

Συντεταγμένες: Λειτουργεί απευθείας με συντεταγμένες pixel, χωρίς να απαιτείται γνώση των παραμέτρων της κάμερας.

Πληροφορία: Περιλαμβάνει τόσο την εξωτερική γεωμετρία (R, t) όσο και τις εσωτερικές παραμέτρους (K) των καμερών.

Βαθμοί Ελευθερίας: Έχει 7 βαθμούς ελευθερίας.

Μαθηματικός Τύπος:

$$E = K'^T F K$$

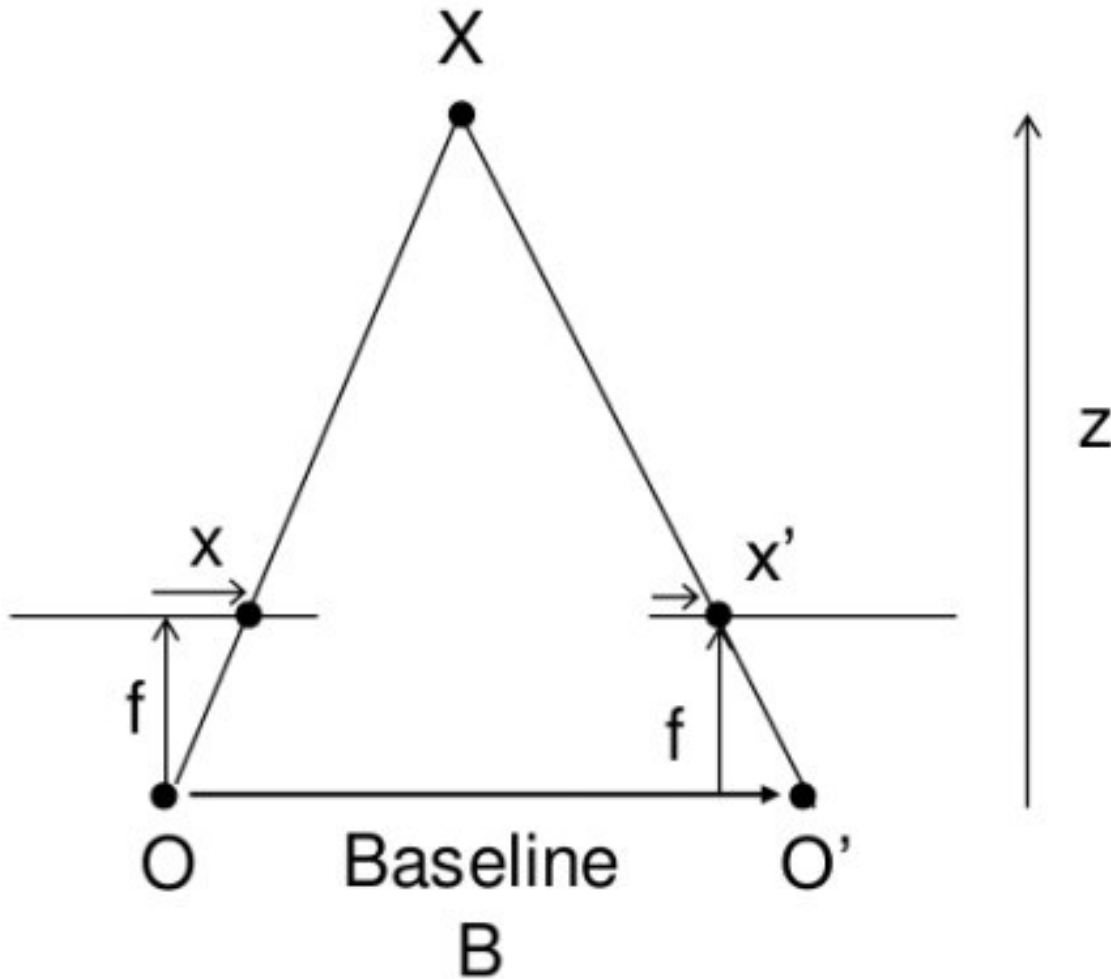
Πρακτική Σημασία

Αν θέλουμε να βρούμε πώς κινήθηκε μια κάμερα στο χώρο (π.χ. σε ένα drone ή ρομπότ), συνήθως υπολογίζουμε πρώτα τον Fundamental Matrix από τα pixel των εικόνων, και αν ξέρουμε το K, τον μετατρέπουμε σε Essential Matrix για να εξάγουμε την περιστροφή και τη μετάθεση (Structure from Motion (SfM) και SLAM)

Stereo: ανομοιότητα (Disparity) και βάθος (Depth)


Παρόμοια τρίγωνα $\rightarrow Z = fB / d$

1) Calibration (K, distortion, baseline B)	2) F/E/H + Rectification	3) Stereo = Matching cost (c)	4) Regularization SGM / SGBM	5) Disparity map $d(x,y)$	6) Depth $Z = fB / d$	7) Point Cloud / Mesh
--	-----------------------------	-------------------------------------	---------------------------------	------------------------------	--------------------------	--------------------------




Βασική σχέση (rectified stereo):

$$d = x - x' = (B \cdot f) / Z \Rightarrow Z = (B \cdot f) / d$$

 B: βάση (baseline) μεταξύ καμερών

 f: εστιακή απόσταση (σε pixels)

 d: ανομοιότητα (disparity) — οριζόντια μετατόπιση

 Z: βάθος (depth) — απόσταση κατά τον άξονα z

Όσο μικραίνει το d (μακρινά σημεία), τόσο αυξάνει η σχετική αβεβαιότητα στο Z.

Επιπολική γεωμετρία (Eipolar geometry)

Γιατί επιταχύνει/περιορίζει το matching

1) Calibration
(K, distortion,
baseline B)

2) F/E/H
+ Rectification

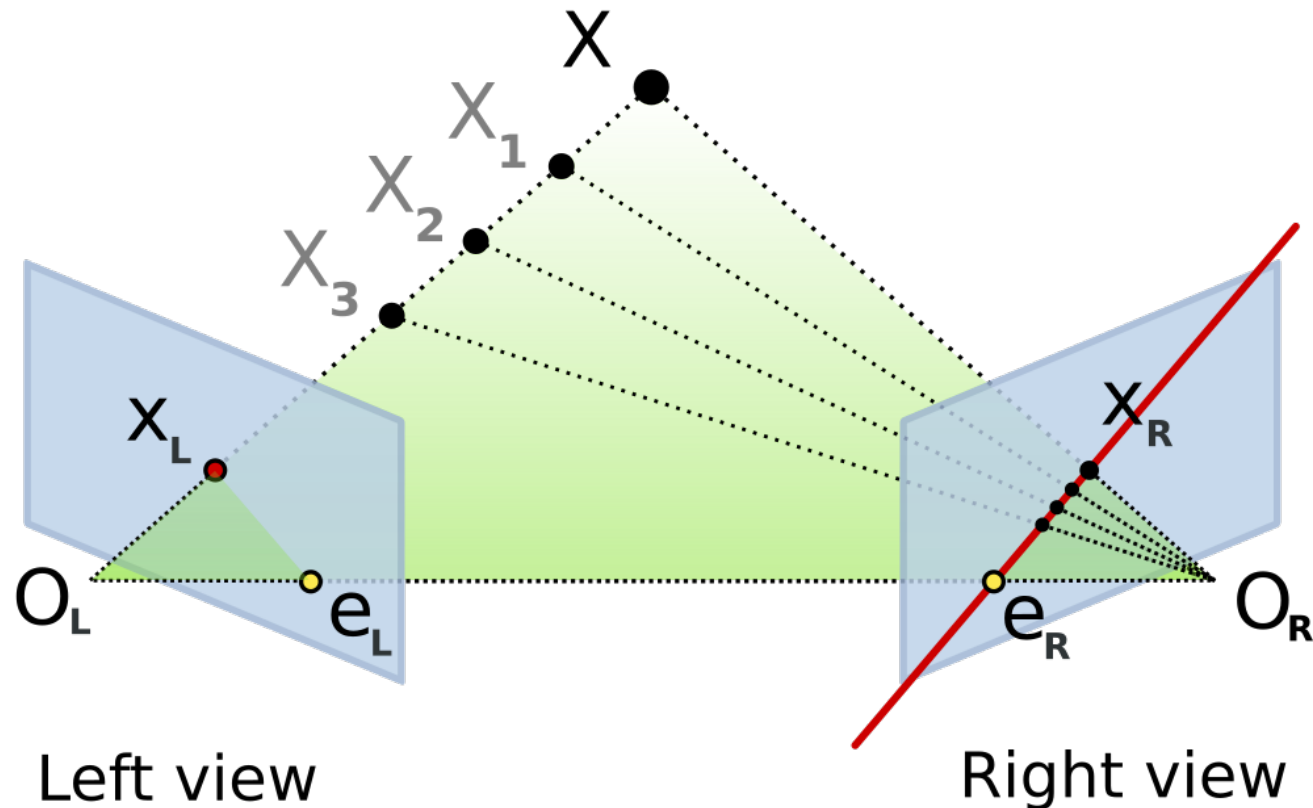
3) Stereo
▶ Matching cost
(C)

4) Regularization
SGM / SGBM

5) Disparity
map $d(x,y)$

6) Depth
 $Z = fb / d$

7) Point Cloud
/ Mesh



Ιδέα:

- ▶ Κάθε σημείο στην αριστερή εικόνα αντιστοιχεί σε επιπολική γραμμή (epipolar line) στη δεξιά.
- ▶ Άρα: αναζητούμε αντιστοίχιση 1D αντί για 2D (πολύ ταχύτερο).
- ▶ Το Fundamental matrix (F) κωδικοποιεί τον περιορισμό $x^T F x = 0$.

Στο rectified stereo, οι επιπολικές γραμμές ευθυγραμμίζονται με τις γραμμές εικόνας → το d γίνεται «καθαρά οριζόντιο».

Stereo rectification

Βάζουμε τις επιπολικές γραμμές «οριζόντιες»

1) Calibration
(K, distortion,
baseline B)

2) F/E/H
+ Rectification

3) Stereo
▶ Matching cost
(C)

4) Regularization
SGM / SGBM

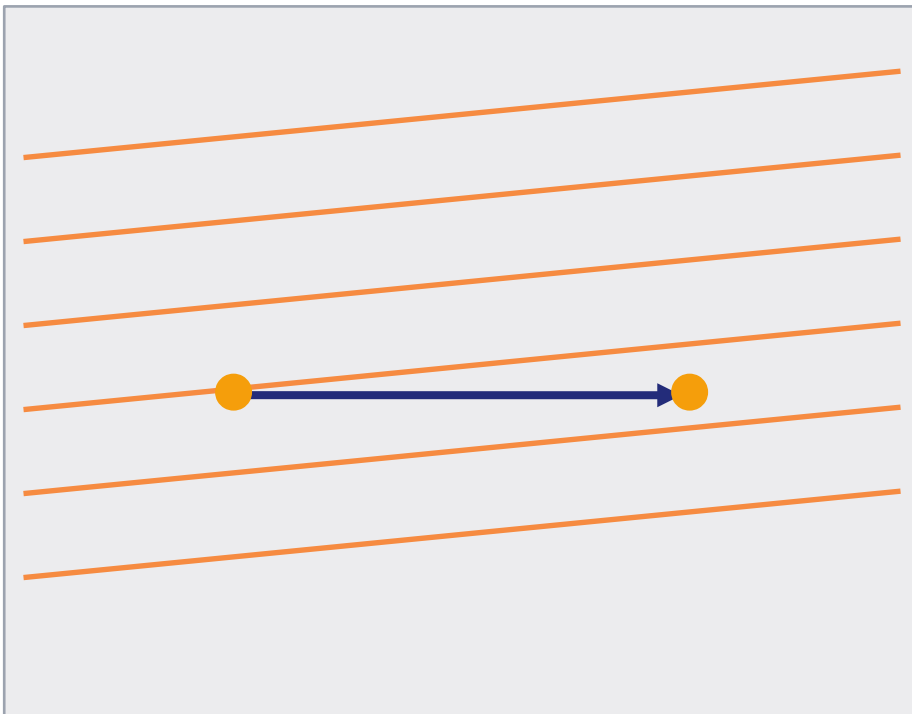
5) Disparity
map $d(x,y)$

6) Depth
 $Z = fB / d$

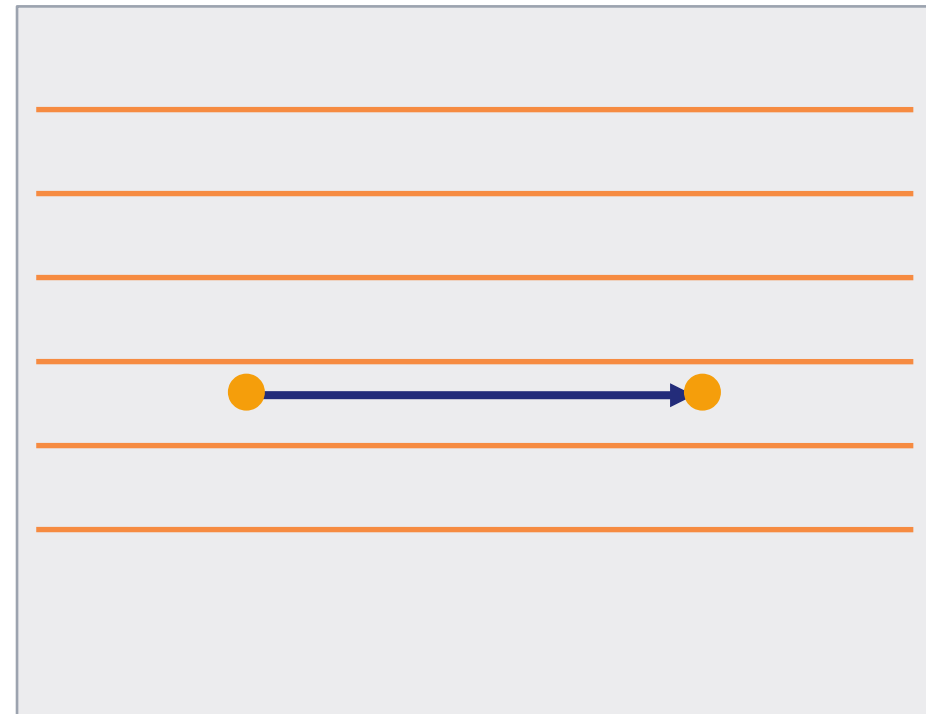
7) Point Cloud
/ Mesh

Στόχος: μετά την ορθοαναγωγή (rectification) οι αντιστοιχίες έχουν ίδιο y (γραμμή) → αναζήτηση μόνο σε x .

Πριν (unrectified)



Μετά (rectified)



Stereo rectification

Βάζουμε τις επιπολικές γραμμές «οριζόντιες»

1) Calibration
(K, distortion,
baseline B)

2) F/E/H
+ Rectification

3) Stereo
Matching cost
(C)

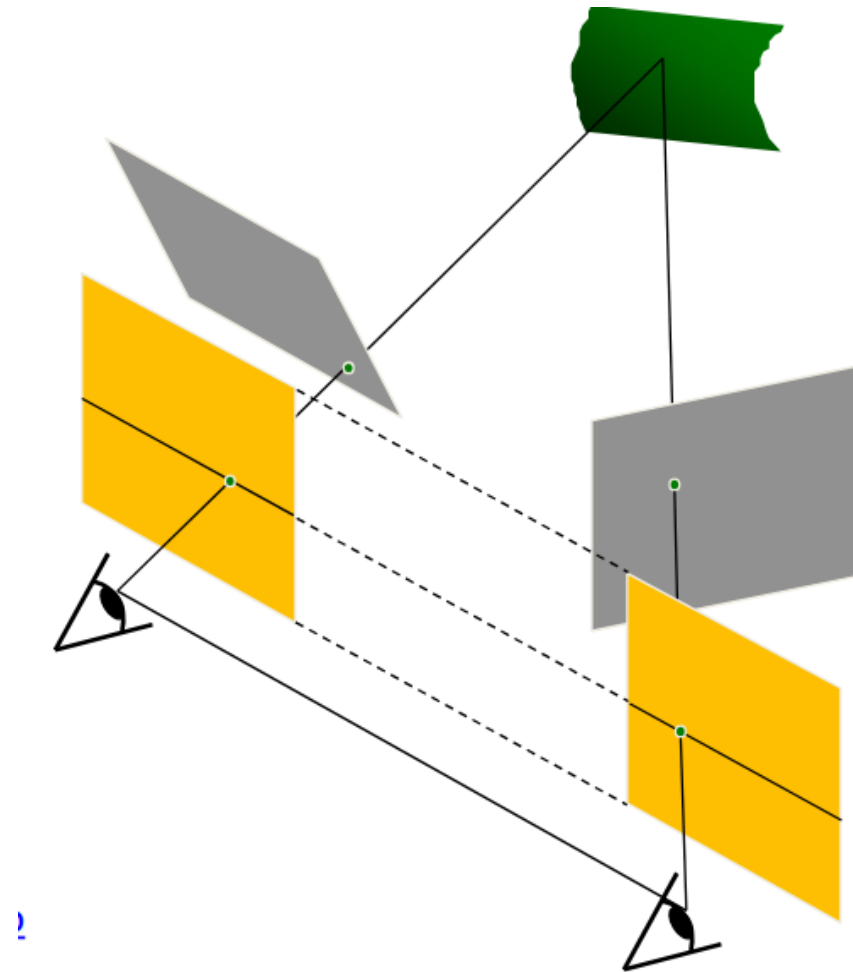
4) Regularization
SGM / SGBM

5) Disparity
map $d(x,y)$

6) Depth
 $Z = fB / d$

7) Point Cloud
/ Mesh

Πριν την ορθοαναγωγή (rectification) οι αντιστοιχίες δεν έχουν ίδιο y (γραμμή) \rightarrow αναζήτηση σε x και y .



Stereo rectification

Βάζουμε τις επιπολικές γραμμές «οριζόντιες»

1) Calibration
(K, distortion,
baseline B)

2) F/E/H
+ Rectification

3) Stereo
Matching cost
(C)

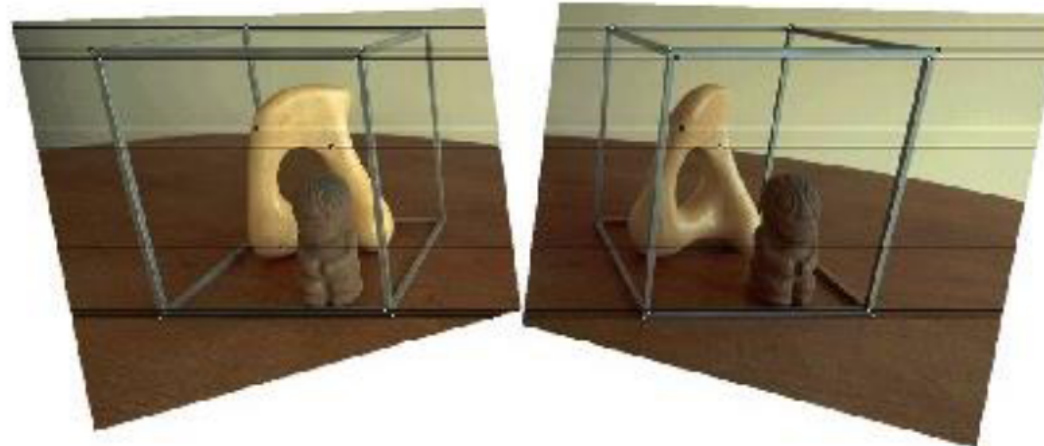
4) Regularization
SGM / SGBM

5) Disparity
map $d(x,y)$

6) Depth
 $Z = fB / d$

7) Point Cloud
/ Mesh

Στόχος: μετά την ορθοαναγωγή (rectification) οι αντιστοιχίες έχουν ίδιο y (γραμμή) \rightarrow αναζήτηση μόνο σε x .



Το πρόβλημα αντιστοίχισης (Correspondence problem)

Γιατί το depth estimation δεν είναι «μία πράξη»

Ο **Fundamental Matrix** μας λέει ότι ένα σημείο στην εικόνα A πρέπει να βρίσκεται πάνω σε μια συγκεκριμένη γραμμή (επιπολική γραμμή) στην εικόνα.



Το Πρόβλημα: "Θόρυβος" στα Δεδομένα



Όταν προσπαθούμε για να βρούμε κοινά σημεία σε δύο φωτογραφίες, πολλά από αυτά τα ζεύγη είναι λανθασμένα (outliers).



Μπορεί δύο σημεία να μοιάζουν οπτικά αλλά να ανήκουν σε διαφορετικά αντικείμενα.



Αν προσπαθήσουμε να υπολογίσουμε τον Θεμελιώδη Πίνακα (Fundamental Matrix) χρησιμοποιώντας όλα τα σημεία μαζί, τα λάθη θα "μολύνουν" τον υπολογισμό και το αποτέλεσμα θα είναι εντελώς άχρηστο. Απαιτείται μια μεθοδολογία εξάλειψης των outliers.

Πρακτική λύση: RANSAC outlier detection and elimination

Το πρόβλημα αντιστοίχισης (Correspondence problem)

Γιατί το depth estimation δεν είναι «μία πράξη»

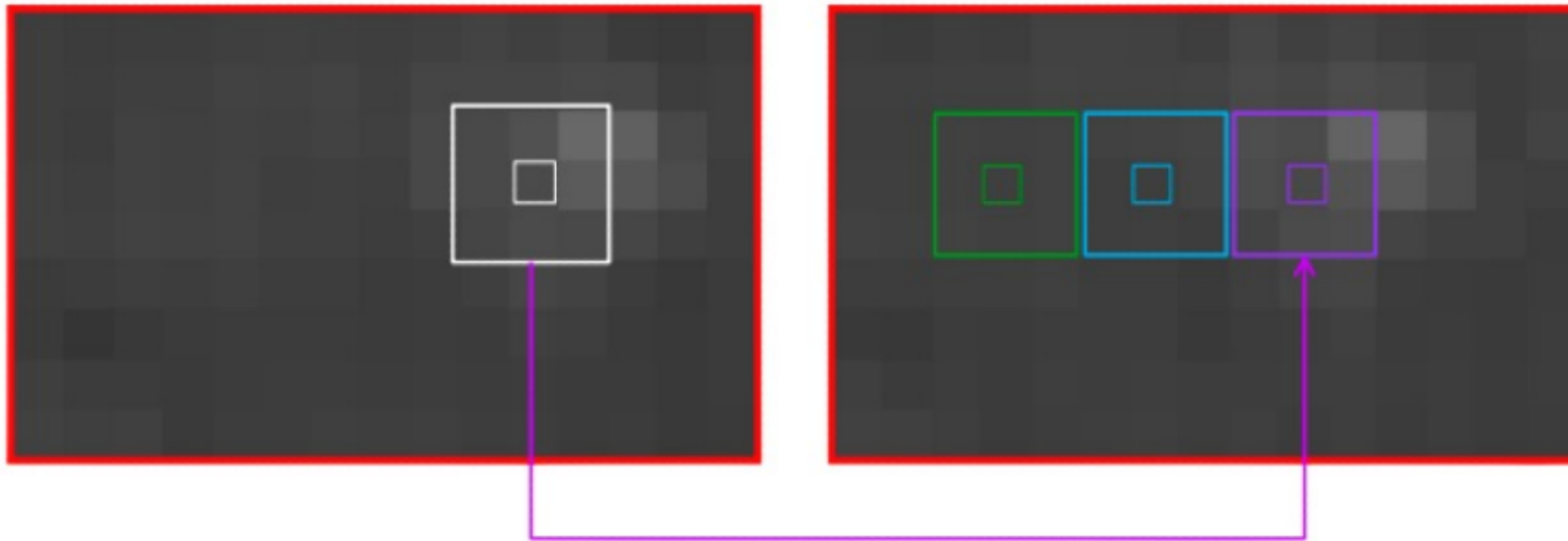


Figure 4: Window Based Correspondence Illustration (Veksler)

Πρακτική λύση: «εύρωστη» (robust) εκτίμηση + ομαλοποίηση (regularization) + έλεγχοι συνέπειας (consistency checks).

Το πρόβλημα αντιστοίχισης (Correspondence problem)

Γιατί το depth estimation δεν είναι «μία πράξη»

- Ομοιόμορφες περιοχές (textureless): πολλά πιθανά matches.
- Επαναλαμβανόμενα μοτίβα (repetitive patterns): λάθος αντιστοιχίσεις.
- Αποκρύψεις (occlusions): σημεία που υπάρχουν μόνο στη μία εικόνα.
- Διαφορές φωτισμού/έκθεσης (radiometric differences): αλλάζει το cost.
- Ανακλάσεις/διαφάνειες (specular/transparent)
- Κινούμενα αντικείμενα (dynamic scenes): δεν ισχύει στατικότητα.

Πρακτική λύση: «εύρωστη» (robust) εκτίμηση + ομαλοποίηση (regularization) + έλεγχοι συνέπειας (consistency checks).

Το πρόβλημα αντιστοίχισης (Correspondence problem)

Γιατί το depth estimation δεν είναι «μία πράξη»

Η λύση: Ο Αλγόριθμος RANSAC

Ο **RANSAC** (Random Sample Consensus) είναι μια επαναληπτική μέθοδος που "μαντεύει" ποια σημεία είναι σωστά. Η διαδικασία λειτουργεί ως εξής:

- 1) Τυχαία Επιλογή: Επιλέγει ένα ελάχιστο δείγμα σημείων (συνήθως 8 σημεία).
- 2) Υπολογισμός: Υπολογίζει έναν υποψήφιο πίνακα F μόνο από αυτά τα 8 σημεία.
- 3) Έλεγχος (Voting): Ελέγχει όλα τα υπόλοιπα σημεία των φωτογραφιών. Πόσα από αυτά "συμφωνούν" με αυτόν τον πίνακα;
- 4) Επανάληψη: Επαναλαμβάνει τα βήματα 1-3 πολλές φορές.
- 5) Επιλογή: Κρατάει τον πίνακα F που είχε τους περισσότερους "υποστηρικτές" (inliers).

Πρακτική λύση: RANSAC outlier detection and elimination

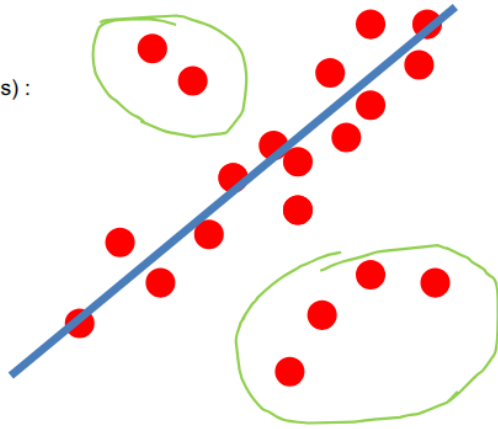
Το πρόβλημα αντιστοίχισης (Correspondence problem)

Γιατί το depth estimation δεν είναι «μία πράξη»

RANSAC

(RANdom SAmples Consensus):

Fischler & Bolles in '81.

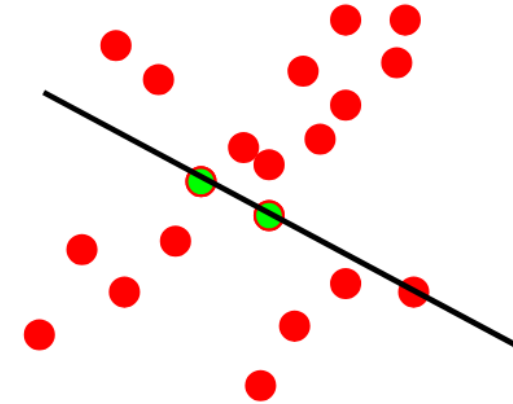


This data is noisy, but we expect a good fit to a known model.

Here, we expect to see a line, but least-squares fitting will produce the wrong result due to strong outlier presence.

RANSAC

Line fitting example



Algorithm:

1. **Sample** (randomly) the number of points required to fit the model ($s=2$)
2. **Solve** for model parameters using samples
3. **Score** by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence

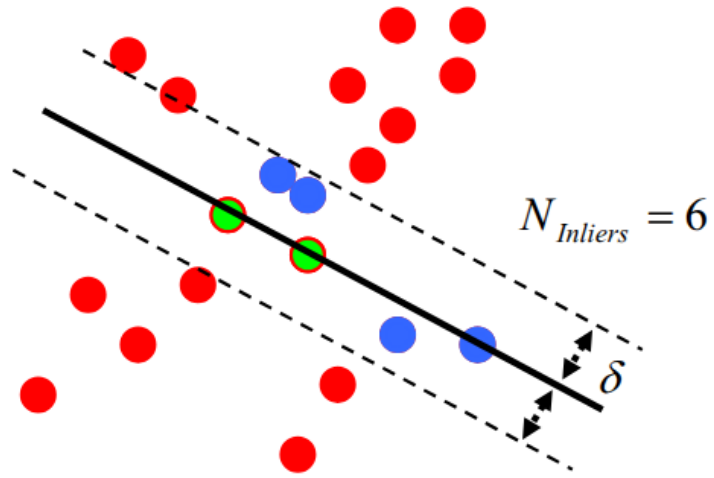
Πρακτική λύση: RANSAC outlier detection and elimination

Το πρόβλημα αντιστοίχισης (Correspondence problem)

Γιατί το depth estimation δεν είναι «μία πράξη»

RANSAC

Line fitting example

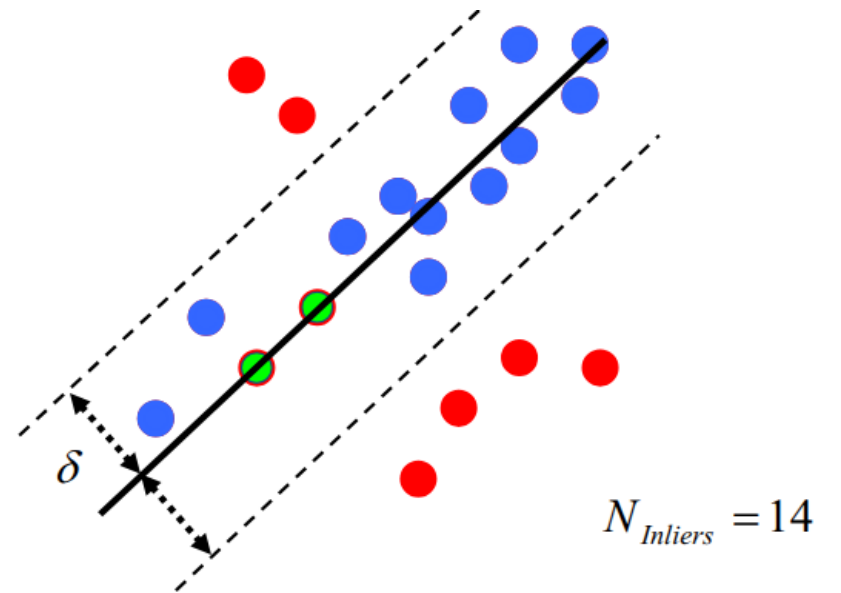


Algorithm:

1. **Sample** (randomly) the number of points required to fit the model ($s=2$)
2. **Solve** for model parameters using samples
3. **Score** by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence

RANSAC



Algorithm:

1. **Sample** (randomly) the number of points required to fit the model ($s=2$)
2. **Solve** for model parameters using samples
3. **Score** by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence

Πρακτική λύση: RANSAC outlier detection and elimination

Πού χρησιμοποιείται RANSAC σε Stereo / 3D Reconstruction

Robust estimation πριν από refinement (least squares / BA)

Stereo (2-view)

- Feature matching → RANSAC fit για F (7/8-point) ή E (5-point)
- Από inliers: rectification + triangulation σταθερότερα
- Homography H (4-point) όταν η σκηνή είναι (σχεδόν) επίπεδη

SfM / Multi-view

- RANSAC PnP: 2D-3D correspondences → pose (inliers)
- Filtering πριν το BA: μειώνει outlier influence
- Τυπικά: RANSAC → non-linear refinement → BA

Stereo matching: συνάρτηση κόστους (Matching cost)

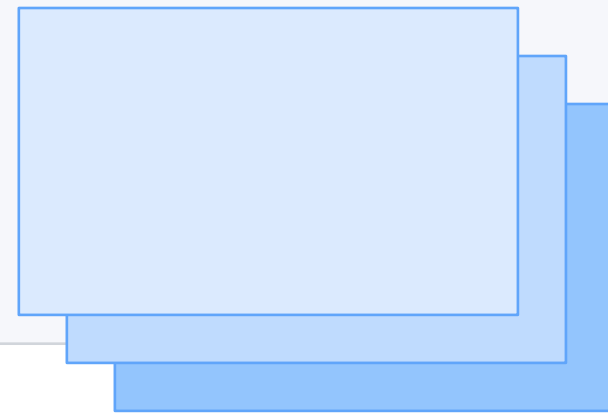
Τι συγκρίνουμε;

Τυπικές επιλογές:

- 🎬 SAD / SSD (Sum of Absolute/Squared Differences) — απλό, ευαίσθητο σε φωτισμό
- 🎬 NCC (Normalized Cross-Correlation) — καλύτερο σε gradients
- 🎬 Census transform (Hamming) — πιο ανθεκτικό (robust) σε φωτισμό
- 🎬 Feature-based matching (SIFT/ORB) — αραιό βάθος (sparse), όχι dense

Κεντρική ιδέα (dense stereo):

Υπολογίζουμε κόστος $C(x,y,d)$ για πολλές πιθανές ανομοιότητες $d \rightarrow$ “cost volume” και μετά βρίσκουμε το καλύτερο d (με/χωρίς ομαλοποίηση).



x,y,d

Stereo matching: συνάρτηση κόστους (Matching cost)

Τι συγκρίνουμε;

Τυπικές επιλογές:

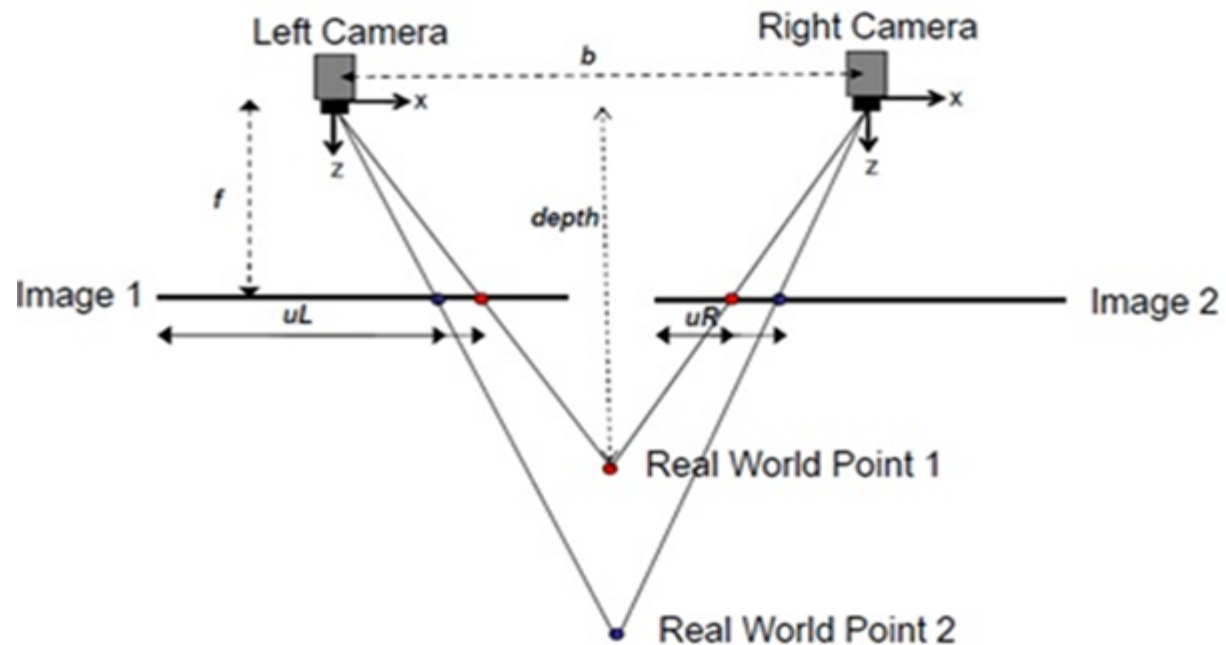
- 🎬 SAD / SSD (Sum of Absolute/Squared Differences) — απλό, ευαίσθητο σε φωτισμό
- 🎬 NCC (Normalized Cross-Correlation) — καλύτερο σε gradients
- 🎬 Census transform (Hamming) — πιο ανθεκτικό (robust) σε φωτισμό
- 🎬 Feature-based matching (SIFT/ORB) — αραιό βάθος (sparse), όχι dense

left image							right image						
3	5	4	4	2	4	2	3	5	4	4	2	4	2
7	4	1	4	4	2	6	7	4	1	4	4	2	6
2	7	46	46	46	6	7	46	46	46	3	6	6	7
5	9	46	46	44	9	7	48	46	44	6	4	9	7
4	7	47	47	47	2	4	47	47	47	7	4	2	4
4	7	56	56	46	6	7	58	56	46	5	6	6	7
3	4	4	1	4	3	2	3	4	4	1	4	3	2

$$\begin{aligned} & (46-48)^2 + (46-46)^2 + (44-44)^2 + \\ & (47-47)^2 + (47-47)^2 + (47-47)^2 + \\ & (56-58)^2 + (56-56)^2 + (46-46)^2 = 8 \end{aligned}$$

Semi-Global Matching (SGM / SGBM)

Συμβιβασμός ποιότητας-χρόνου



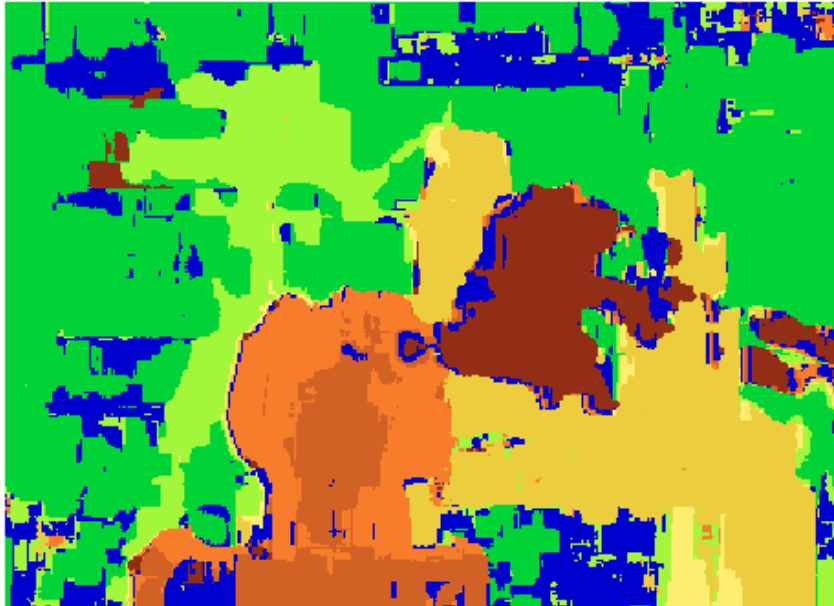
Τι κάνει (high level):

- 🎬 Συσσωρεύει κόστος κατά μήκος πολλών κατευθύνσεων (paths)
- 🎬 Επιβάλλει ήπια ομαλότητα (smoothness) χωρίς πλήρη global optimization
- 🎬 Δουλεύει καλά σε πραγματικό χρόνο (real-time) με σωστή υλοποίηση

Το OpenCV υλοποιεί SGBM (StereoSGBM) ως πρακτική παραλλαγή του SGM για dense disparity.

Παράδειγμα: disparity map (OpenCV)

Κλασικό output από block matching



Window-based matching
(best window size)



‘Ground truth’

Σημείωση: το αποτέλεσμα μπορεί να είναι θορυβώδες → χρειάζονται σωστές παράμετροι + post-filtering.

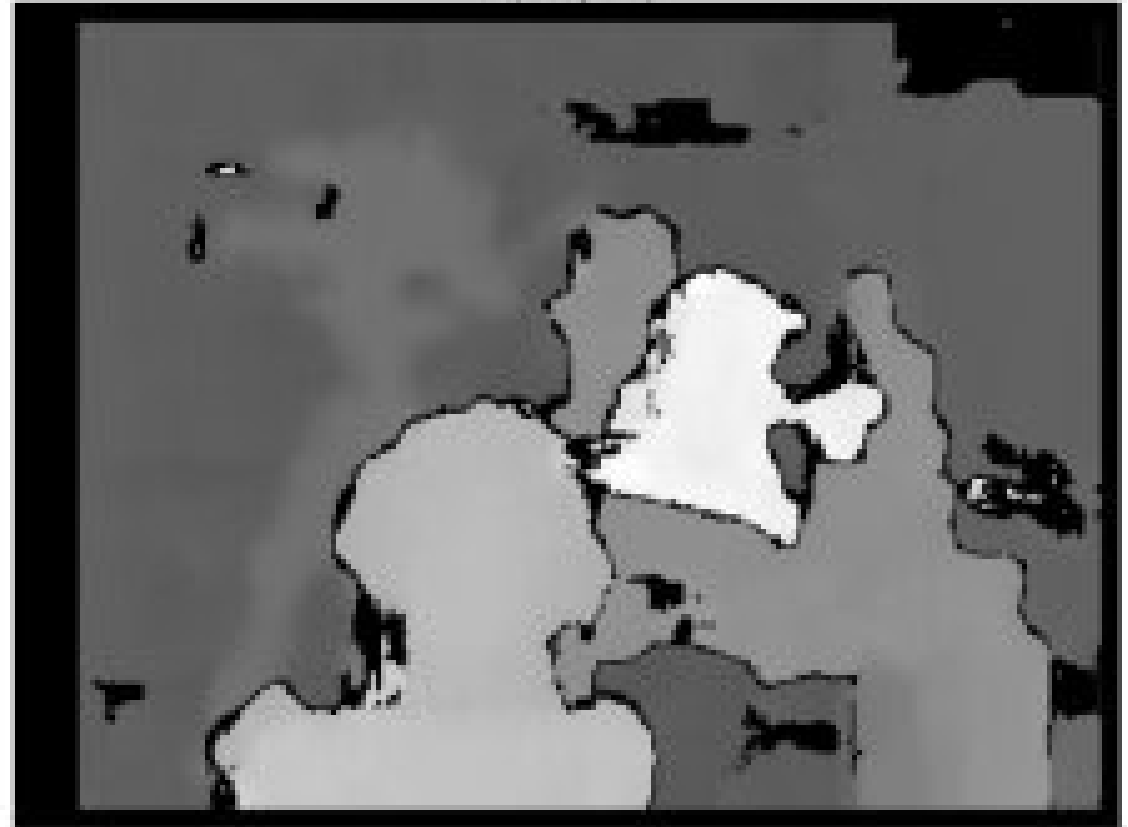
Παράδειγμα: disparity map (OpenCV)

Κλασικό output από block matching

Original image



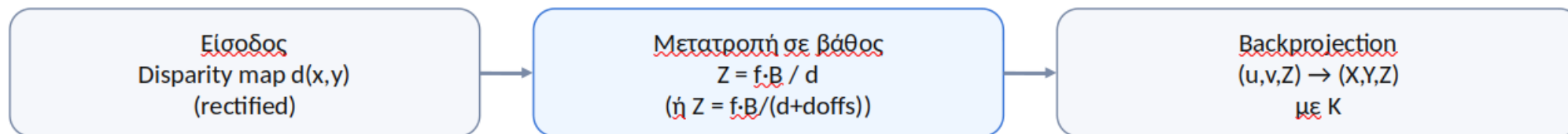
Disparity Map



Σημείωση: το αποτέλεσμα μπορεί να είναι θορυβώδες → χρειάζονται σωστές παράμετροι + post-filtering.

3D Backprojection: Από disparity σε metric depth

Rectified stereo $\rightarrow d(x,y) \rightarrow Z(x,y) \rightarrow (X,Y,Z)$



Τύποι (pinhole, intrinsics K):

$$Z = fB / d$$

$$X = (u - cx) \cdot Z / fx$$

$$Y = (v - cy) \cdot Z / fy$$

Σημ.: u,v είναι pixels της rectified εικόνας (μετά από undistort+rectify).

Links:

https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html (Q, reprojectImageTo3D)

https://docs.opencv.org/4.x/dd/d53/tutorial_py_depthmap.html

Μετα-επεξεργασία (Post-processing)

Για καθαρότερο disparity → καλύτερο depth

- 🎬 Speckle filtering: αφαίρεση μικρών «νησίδων» (speckles).
- 🎬 Left-right consistency check: $d_L(x,y) \approx d_R(x-d,y)$.
- 🎬 Sub-pixel refinement: βελτίωση ακρίβειας d (π.χ. παρεμβολή).
- 🎬 Edge-aware filtering: διατήρηση ακμών (discontinuities) στο βάθος.
- 🎬 Hole filling / inpainting: συμπλήρωση occlusions με προσοχή.

Πρακτικό: συχνά βελτιώνει το αποτέλεσμα περισσότερο με post-processing παρά με «μαγικές» αλλαγές στο matcher.




Depth from disparity: απο τα pixels στα μέτρα

$Z = (B \cdot f) / d$ και πρακτικές μονάδες

$$Z = (B \cdot f) / d$$

f σε pixels • B σε μέτρα • d σε pixels \Rightarrow Z σε μέτρα

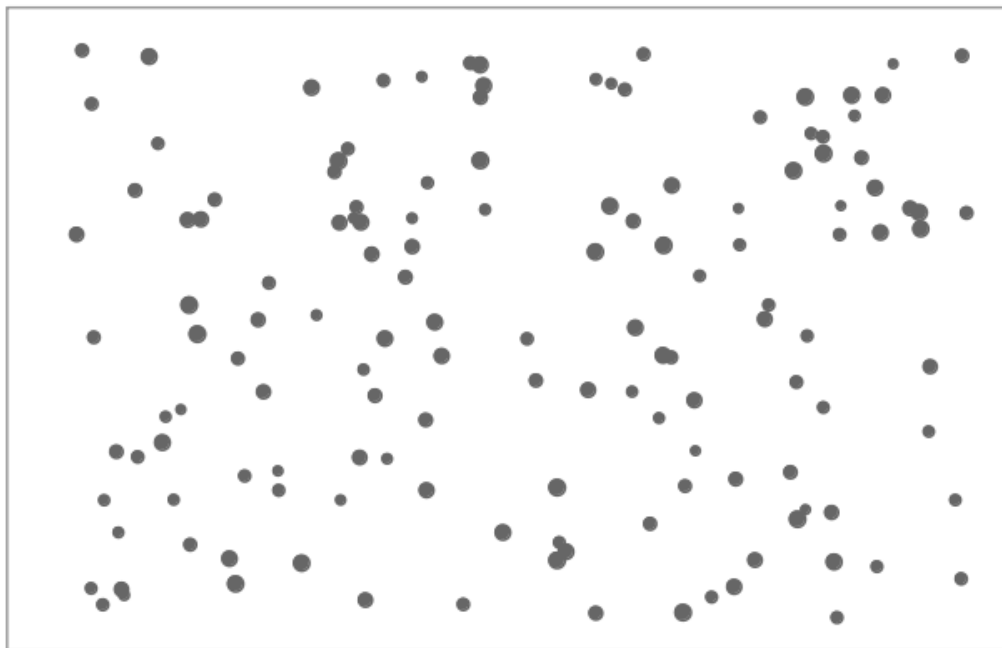
Συχνά λάθη:

-  Χρήση λάθος f (π.χ. mm αντί για pixels).
-  Baseline B λάθος/άγνωστο \rightarrow λάθος κλίμακα.
-  Rectification λάθος \rightarrow το d δεν είναι αμιγώς οριζόντιο.

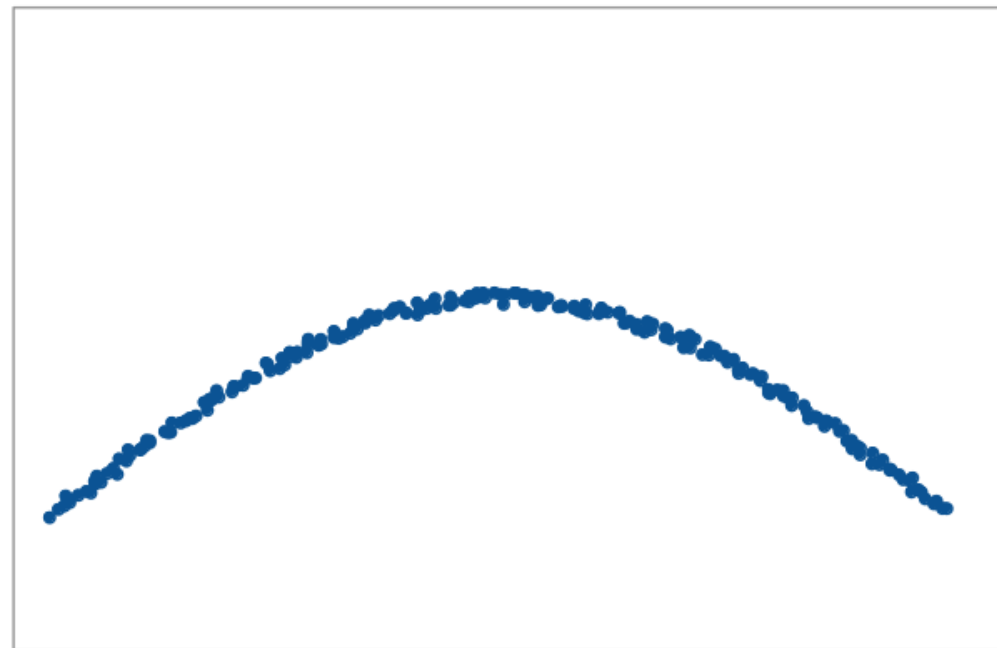
Point Cloud: κατασκευή + καθαρισμός

Depth/3D χωρίς quality control → θόρυβος, speckles, λάθος points

Raw disparity/depth → noisy 3D



Filtered disparity/depth → cleaner 3D

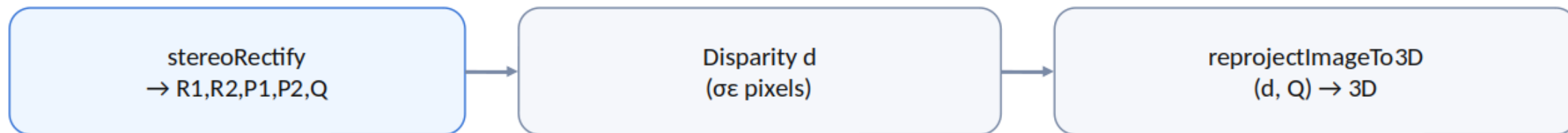


Συνήθη βήματα post-processing:

- Left-right consistency check → occlusions mask
- Speckle filtering, median/bilateral/WLS filtering (edge-aware)
- Confidence thresholding (cost curve sharpness / uniqueness ratio)
- Depth range clipping + voxel downsample (για visualization)

Q (Reprojection Matrix): γρήγορος τρόπος για 3D

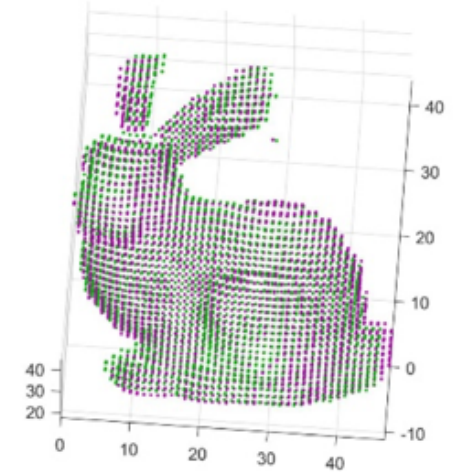
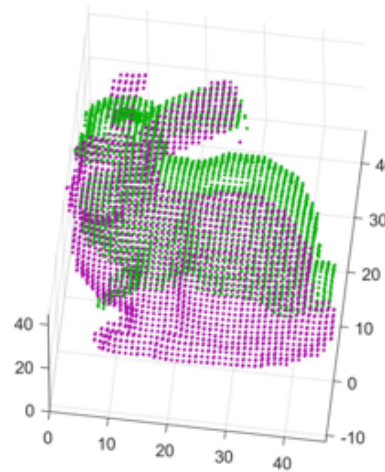
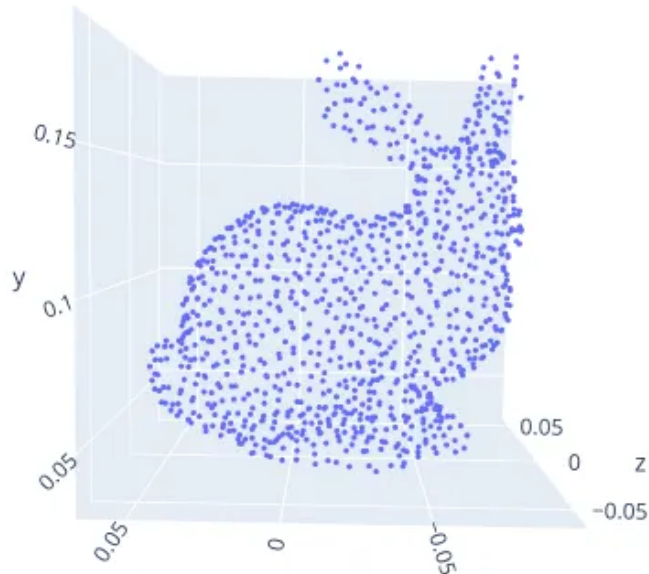
Output του stereoRectify → απευθείας disparity → 3D



Τι κάνει το Q;

- Κωδικοποιεί τη γεωμετρία rectified stereo (baseline, focal, principal points).
- Δίνει 3D σημείο για κάθε pixel: $(X,Y,Z,W) = Q \cdot (u,v,d,1)^T$, μετά divide by W.
- Πρακτικό όταν έχουμε calibrated stereo και θέλουμε άμεση δημιουργία point cloud.

Backprojection – ICP algorithm



Πώς συνεργάζονται (ICP + Backprojection)

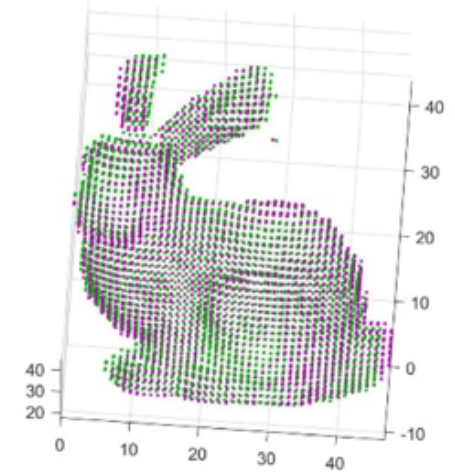
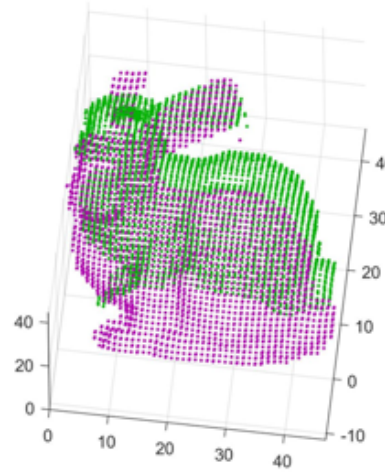
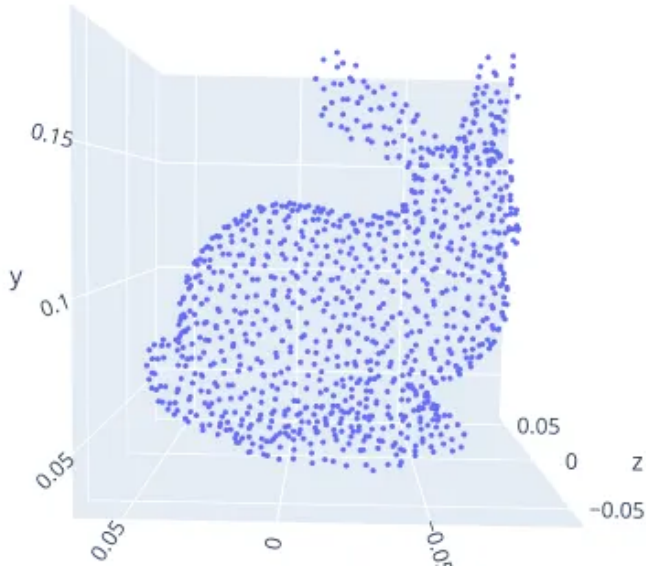
Σε ένα σύστημα Stereo Tracking ή SLAM:

Backprojection: Η στερεοσκοπική κάμερα λαμβάνει δύο εικόνες, υπολογίζει το βάθος και παράγει ένα 3D νέφος σημείων για το τρέχον καρέ.

ICP Alignment: Ο αλγόριθμος ICP συγκρίνει το νέο αυτό νέφος με το νέφος του προηγούμενου καρέ (ή με ένα προϋπάρχον 3D μοντέλο).

Pose Estimation: Η μεταβολή που υπολογίζει ο ICP δείχνει ακριβώς πώς μετακινήθηκε η κάμερα ή το αντικείμενο στον χώρο (6 βαθμοί ελευθερίας - 6-DOF).

Backprojection – ICP algorithm



Τι είναι το Backprojection (Οπισθοπροβολή);

Στη στερεοσκοπική όραση, το Backprojection είναι η διαδικασία μετατροπής των δισδιάστατων (2D) εικονοστοιχείων των εικόνων σε τρισδιάστατα (3D) σημεία στον χώρο.

Πώς λειτουργεί: Χρησιμοποιώντας τις εσωτερικές παραμέτρους της κάμερας (εστιακή απόσταση, οπτικό κέντρο) και τη διαφορά φάσης (disparity) μεταξύ της αριστερής και δεξιάς εικόνας, κάθε pixel προβάλλεται πίσω στον 3D χώρο ως μια ακτίνα.

Αποτέλεσμα: Δημιουργείται ένα νέφος σημείων (**point cloud**), το οποίο αναπαριστά τη γεωμετρία της σκηνής.

Ο **ICP** είναι ένας επαναληπτικός αλγόριθμος που χρησιμοποιείται για την ευθυγράμμιση (registration) δύο διαφορετικών νεφών σημείων.

Η διαδικασία:

Αντιστοίχιση: Βρίσκει τα πλησιέστερα σημεία μεταξύ ενός "πηγαίου" νέφους και ενός νέφους "στόχου".

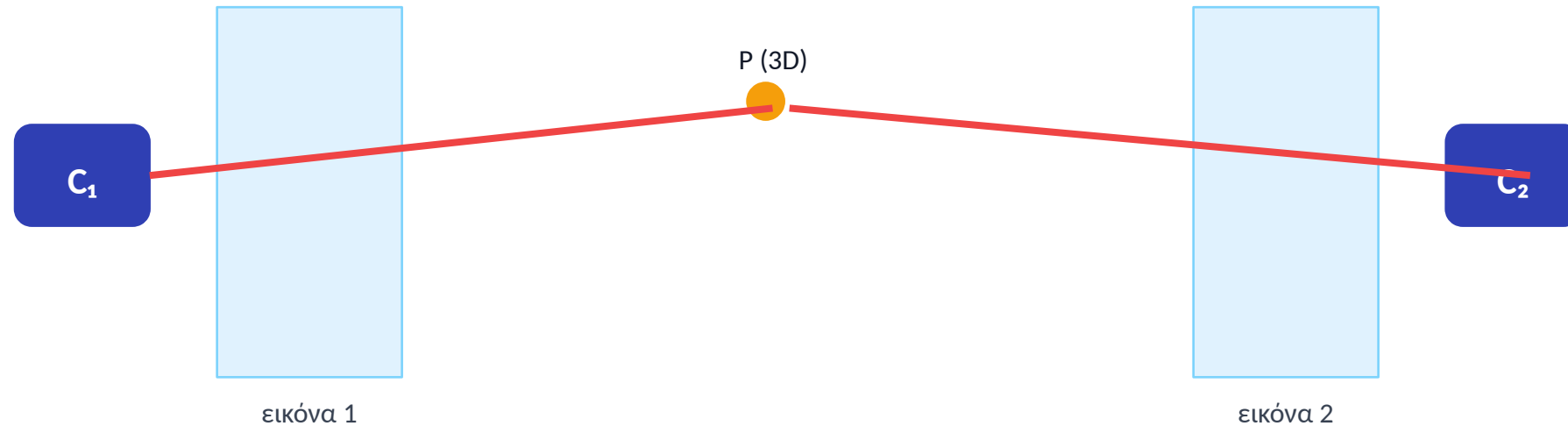
Ελαχιστοποίηση: Υπολογίζει τη βέλτιστη μετατόπιση και περιστροφή που ελαχιστοποιεί την απόσταση μεταξύ των αντιστοιχισμένων σημείων.

Επανάληψη: Επαναλαμβάνει τα βήματα μέχρι τα δύο νέφη να "κουμπώσουν" με ακρίβεια.

Τριγωνισμός (Triangulation)

Πώς «γεννιέται» το 3D από 2 προβολές

Τομή ακτίνων (rays) από δύο κάμερες → εκτίμηση 3D σημείου

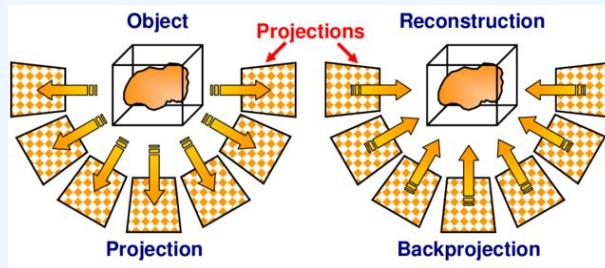


- 🎬 Σε πραγματικές συνθήκες, οι ακτίνες δεν τέμνονται ακριβώς (θόρυβος) → λύση ελάχιστων τετραγώνων (least squares).
- 🎬 Με περισσότερες όψεις (multi-view) το πρόβλημα γίνεται καλύτερα ορισμένο → SfM + Bundle Adjustment (BA).

Backprojection vs Triangulation (και πότε τι)

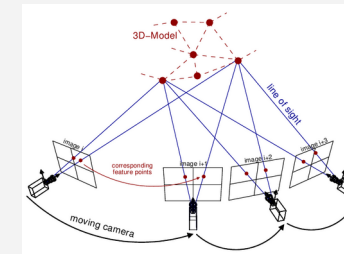
Dense stereo (pixelwise) vs sparse multi-view geometry

Backprojection (dense stereo)



- Ξεκινά από disparity map $d(x,y)$ σε rectified εικόνες.
- Μετατρέπει κάθε pixel σε (X,Y,Z) → πυκνό point cloud.
- Απαιτεί calibration (K,B) για metric scale.
- Ιδανικό για real-time robotics / driving scenarios.

Triangulation (sparse / SfM)



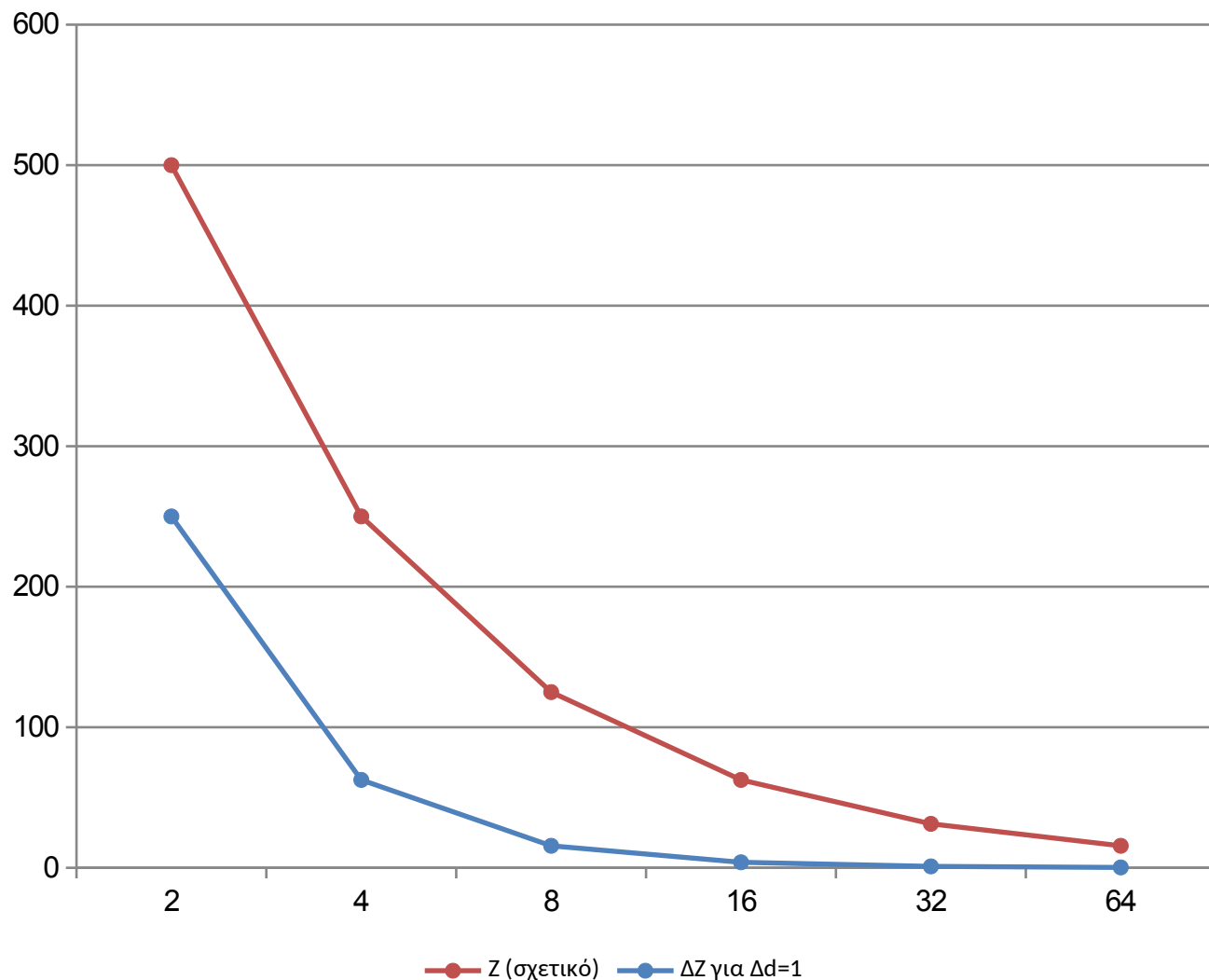
- Ξεκινά από feature matches σε πολλές όψεις (tracks).
- Υπολογίζει camera poses (E/PnP) + triangulation σημείων.
- Βελτιστοποιεί με Bundle Adjustment (reprojection error).
- Scale ambiguity χωρίς πρόσθετη πληροφορία (stereo/IMU/GCP).

Rule of thumb: Dense stereo → disparity→backprojection. Multi-view mapping → SfM (poses+triangulation) → (optional) dense MVS.

Γιατί το βάθος χάνει ακρίβεια μακριά;

Αβεβαιότητα (uncertainty) από το d

Αν το disparity έχει σφάλμα Δd , τότε περίπου: $\Delta Z \approx (B \cdot f / d^2) \cdot \Delta d$



Ερμηνεία:

- 🎬 Για μικρό d (μακριά), το Z μεγαλώνει γρήγορα.
- 🎬 Το ίδιο Δd προκαλεί πολύ μεγαλύτερο ΔZ .
- 🎬 Άρα: το stereo είναι «καλύτερο» σε κοντινές αποστάσεις.

Αυτό εξηγεί γιατί σε driving datasets το stereo δυσκολεύεται σε πολύ μακρινούς στόχους.

Τι κάνει κάθε στάδιο

Inputs/outputs + πρακτικές παγίδες (failure modes)

Στάδια (inputs → outputs)

- Calibration: checkerboard/rig → K , distortion, baseline B
- Undistort: raw images → διορθωμένες εικόνες
- Rectification: $F/E (+K, R, t)$ → οι επιτολικές ευθείες γίνονται οριζόντιες
- Matching cost: (L, R) → $C(x, y, d)$ (π.χ. SAD, Census, NCC)
- Regularization (SGM): C → συνεπές disparity (τιμωρία ασυνέχειας)
- Disparity → Depth: $Z = fB/d$ (ή $Z = fB/(d+d_{offs})$ σε datasets)
- Depth → 3D: back-projection → point cloud / mesh

Κρίσιμες παγίδες

- Λάθος calibration → λάθος depth κλίμακα (συστηματικό σφάλμα)
- Occlusions → “τρύπες” στο disparity (L-R consistency check)
- Textureless περιοχές → ασαφές minimum στο $C(d)$
- Repetitive patterns → λάθος αντιστοιχίσεις (ambiguity)
- Specular/lighting changes → mismatch (χρήσιμα: Census/robust costs)
- Sub-pixel disparity → καλύτερο Z (αλλά ευαίσθητο σε θόρυβο)
- Depth error grows with distance: $Z \sim 1/d$ (μακριά χειροτερεύει)

Structured Lighting

3D αναπαράσταση για downstream tasks

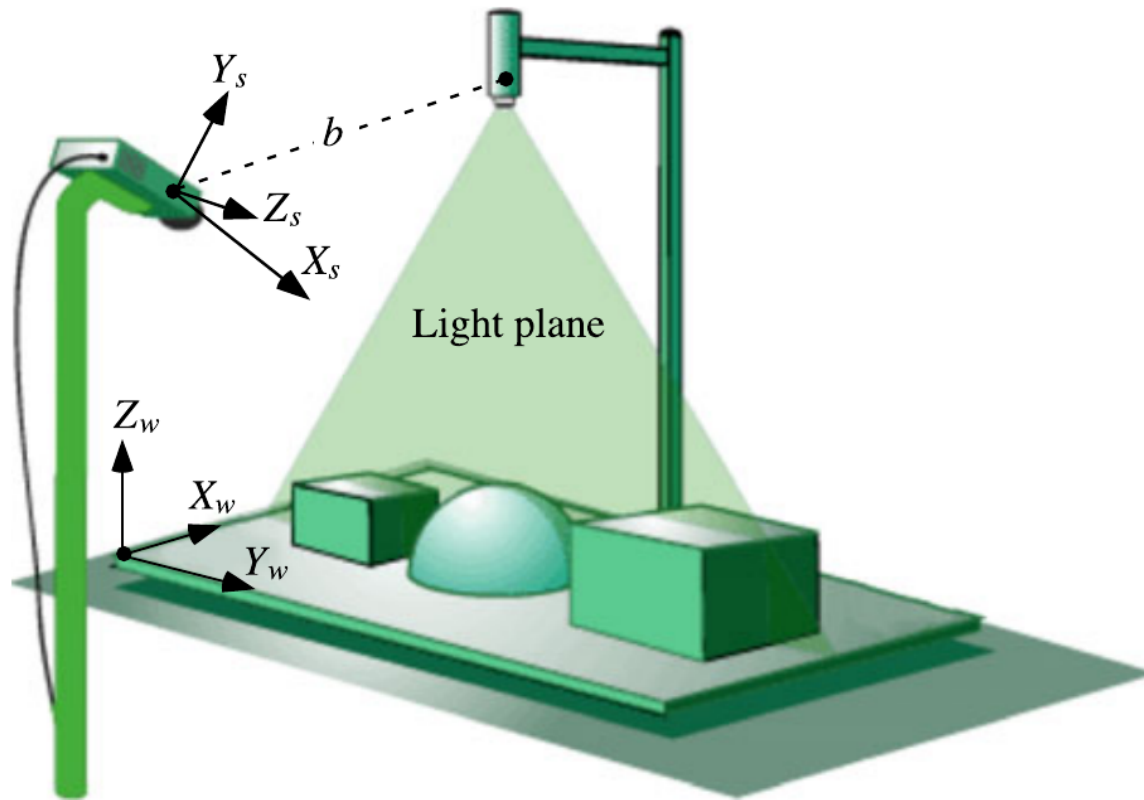
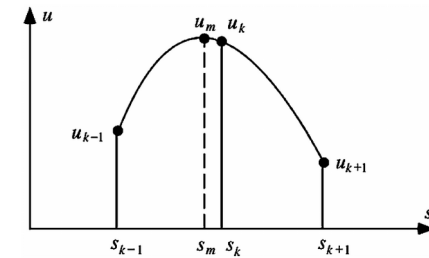


Fig. 7.9 *Left:* The (laser) light source projects a sweeping light plane (by fanning out a light beam into a “sheet of light”), which projects at each time a bright line across the objects of interest. *Right:* The recovered surface of a person using structured lighting



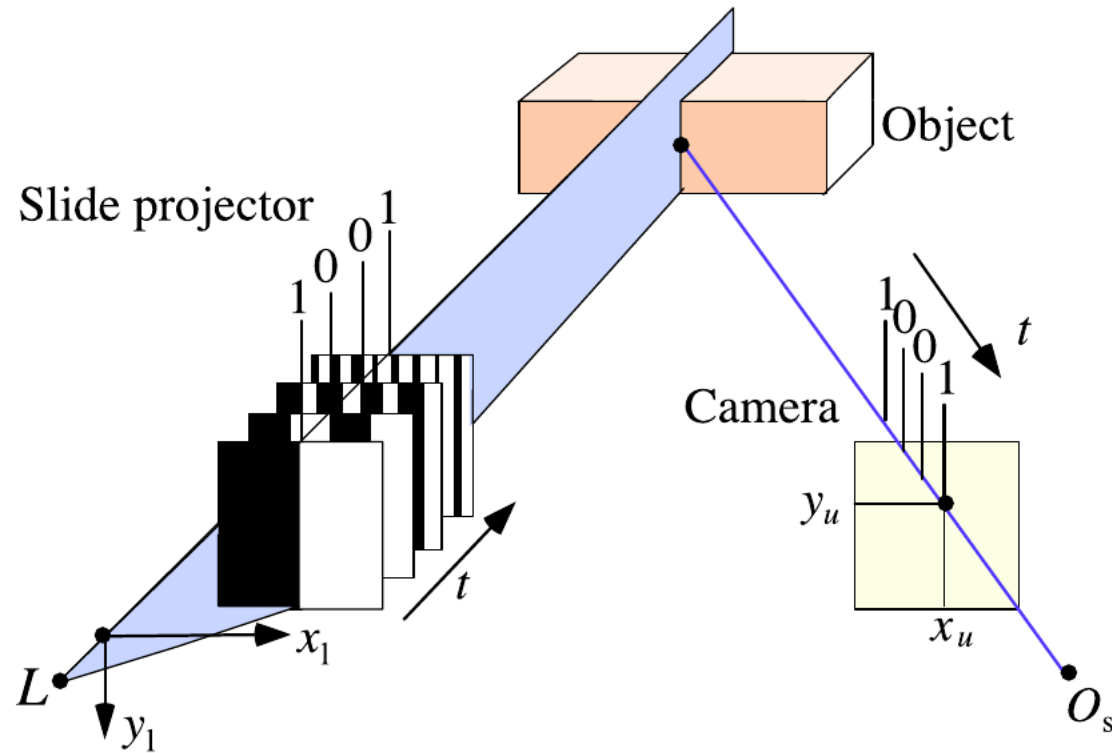


Fig. 7.11 A slide projector projects one binary pattern at a time, thus either illuminating surface points at this moment through the white columns or not (i.e. light is blocked by the *black columns*). A sequence of slides generates thus at a surface point a binary code (1 = illuminated by this slide, 0 = not illuminated). The figure illustrates one light plane defined by subsequent columns in the used binary patterns

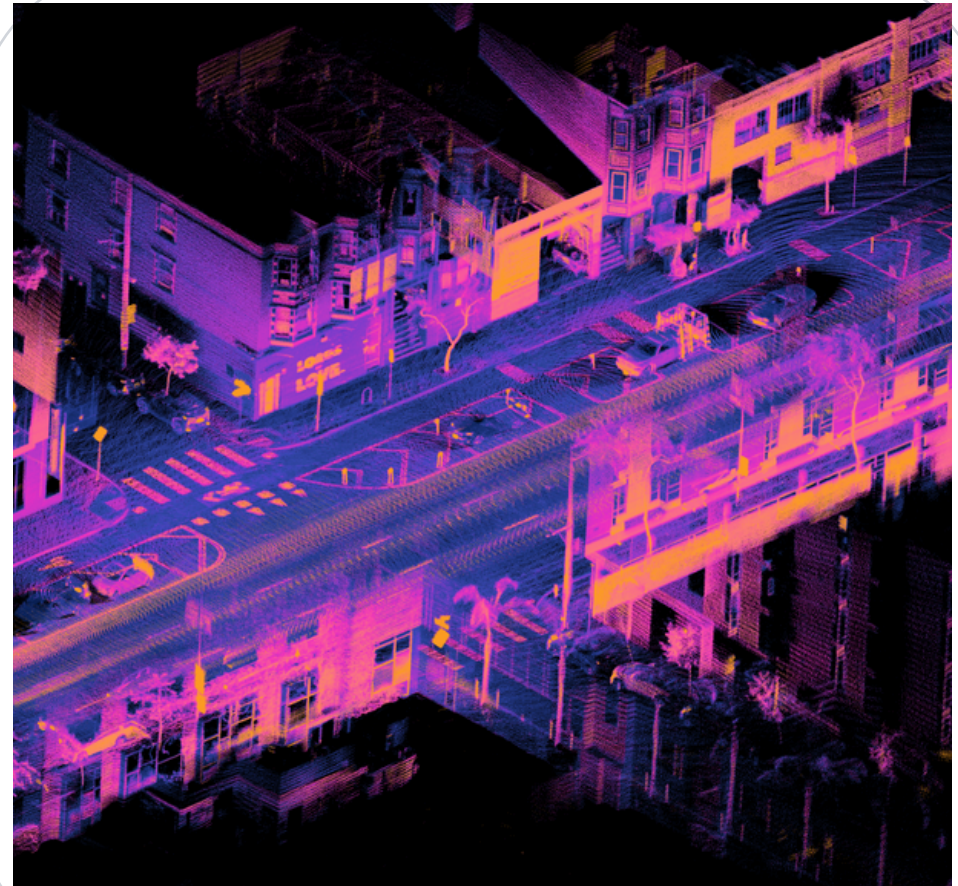
Από depth map σε νέφος σημείων (Point cloud)

3D αναπαράσταση για downstream tasks

Με γνωστό K (intrinsics) μετατρέπουμε $(u,v,Z) \rightarrow (X,Y,Z)$ με back-projection.



Παράδειγμα: ToF point cloud



Παράδειγμα: LiDAR point cloud

Structure from multiple image views

3D αναπαράσταση για downstream tasks



Structure from multiple image views

3D αναπαράσταση για downstream tasks



SfM pipeline (Structure from Motion)

Από πολλές εικόνες σε sparse 3D + camera poses

Στον πυρήνα: αντιστοιχίσεις (matches) + εύρωστη εκτίμηση (robust) + τριγωνισμός + βελτιστοποίηση.



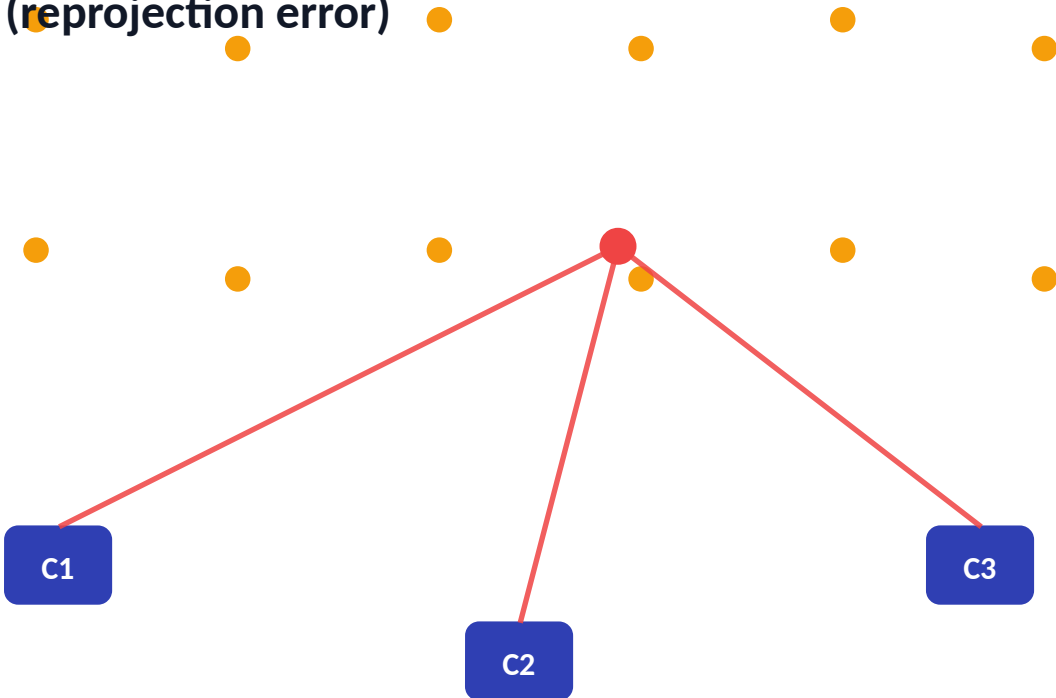
Σημείο-κλειδί: η εύρωστη εκτίμηση (robust estimation) απορρίπτει outliers πριν «μολύνουν» τη γεωμετρία.

Bundle Adjustment (BA)

Βελτιστοποίηση (non-linear least squares)

BA: βρίσκουμε τις παραμέτρους καμερών και 3D σημείων που ελαχιστοποιούν το σφάλμα επαναπροβολής (reprojection error).

Ελαχιστοποίηση σφάλματος επαναπροβολής (reprojection error)



Γιατί είναι κρίσιμο;

- 🎬 Διορθώνει σταδιακά drift και μικρά σφάλματα γεωμετρίας.
- 🎬 Μειώνει τον θόρυβο στη 3D ανακατασκευή.
- 🎬 Βασικό bottleneck σε χρόνο/μνήμη για μεγάλα datasets.

Πρακτικά: BA = ο «κινητήρας ποιότητας» του SfM.

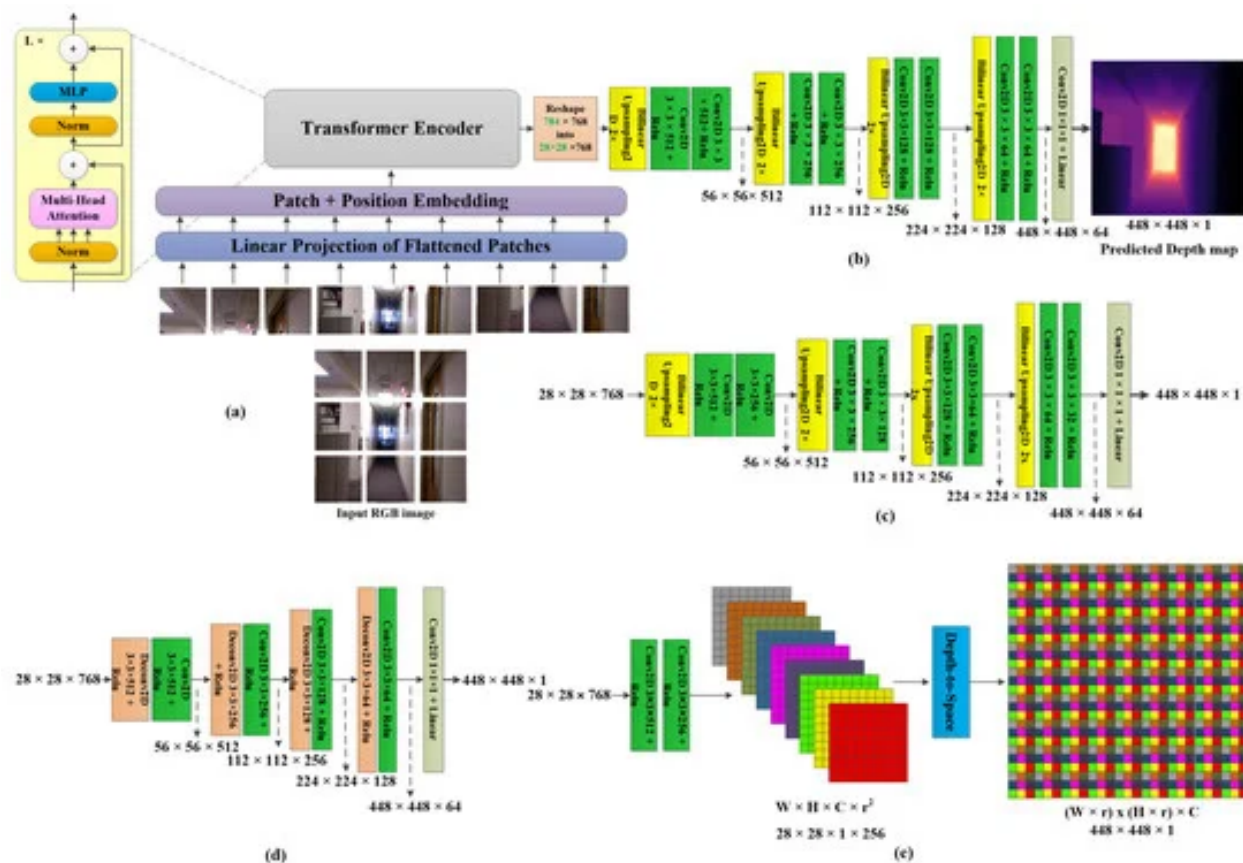
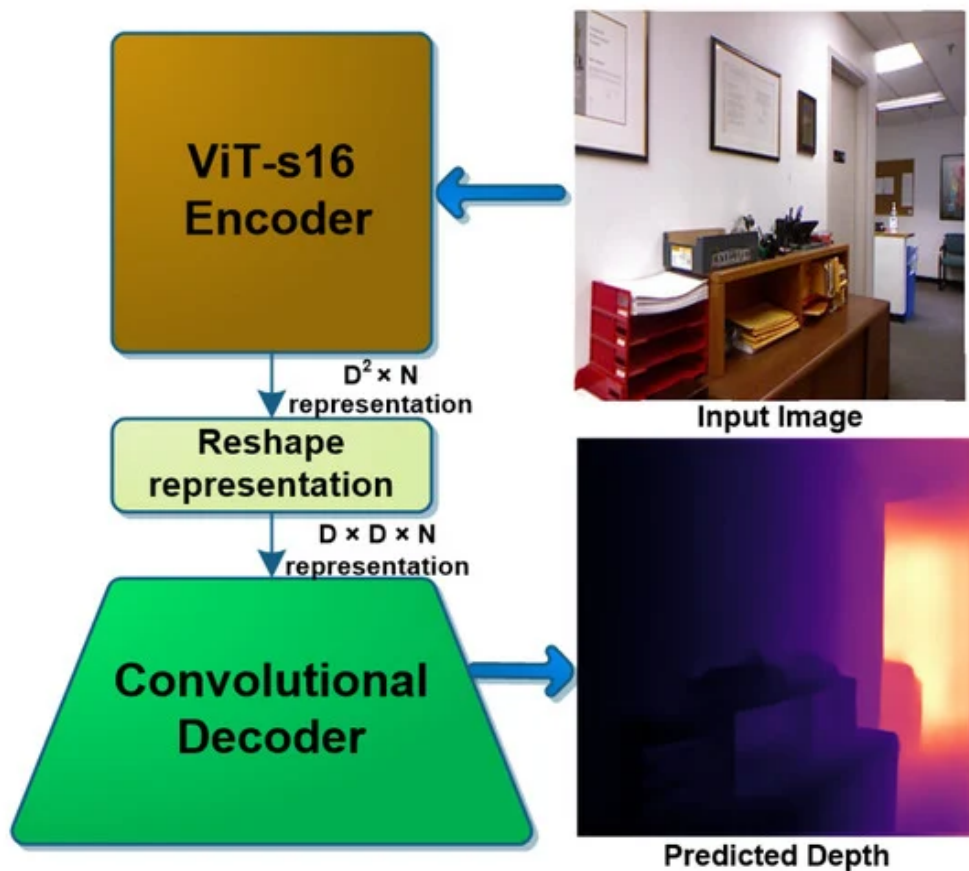
Stereo vs SfM vs MVS

Πότε επιλέγω τι;

Κριτήριο	Stereo	SfM	MVS
Input	2 εικόνες (ή stream)	N επικαλυπτόμενες εικόνες	N εικόνες + (συχνά) poses
Output	Dense disparity/depth	Sparse 3D + poses	Dense 3D / mesh
Κλίμακα	Γνωστή (baseline)	Αμφίσημη (scale ambiguity)	Εξαρτάται από poses/scale
Καλό για	Real-time, robotics	Mapping/phototourism	High-quality models
Κόστος	Μεσαίο (SGM)	Υψηλό (BA)	Πολύ υψηλό

Learning-based depth estimation

3D αναπαράσταση για downstream tasks



RT-ViT: Real-Time Monocular Depth Estimation Using Lightweight Vision Transformers

Monocular depth estimation: εκτιμά βάθος από μία εικόνα (single image).

Τι κερδίζουμε;

- 🎬 Δουλεύει με 1 κάμερα (φθηνότερο HW)
- 🎬 Καλό για “relative depth” και scene understanding
- 🎬 Συνδυάζεται με stereo/LiDAR για depth completion

Τι πληρώνουμε;

- 🎬 Απόλυτη κλίμακα (metric depth) δύσκολη χωρίς επιπλέον cues
- 🎬 Γενίκευση (generalization) εκτός dataset: θέλει προσοχή

Παραδείγματα (links):

- MiDaS: robust monocular depth (zero-shot transfer)
- KITTI depth prediction/completion benchmarks

Σημαίνει: “monocular \neq δωρεάν metric depth”.

Το βλέπουμε ως συμπληρωματικό εργαλείο (fusion) με stereo/LiDAR.

Δημοφιλή benchmarks:

- 🎬 KITTI Stereo (2012/2015): outdoor driving, disparity GT από LiDAR/SLAM
- 🎬 Middlebury Stereo: indoor/high-res, κλασική αξιολόγηση dense stereo
- 🎬 ETH3D: stereo + multi-view stereo / 3D reconstruction, laser-scanned GT

Μετρικές (metrics) — ενδεικτικά:

- 🎬 EPE (End-Point Error) / MAE στο disparity
- 🎬 Bad pixel % (π.χ. $> 3\text{px}$ ή $> 5\%$)
- 🎬 RMSE / AbsRel στο depth (όταν υπάρχει metric GT)
- 🎬 Completeness vs accuracy (ιδίως σε MVS)

Σημαντικό: συγκρίνεις “ίδιες” συνθήκες (rectification, masks, ranges). Δεν συγκρίνεις απευθείας disparity-metrics με depth-metrics χωρίς προσοχή.

- 🎬 Λάθος/ασταθές calibration → λάθος βάθος παντού.
- 🎬 Μικρό baseline B → μικρά disparities → χειρότερη ακρίβεια σε μέση/μακρινή ζώνη.
- 🎬 Μη συγχρονισμένες κάμερες (rolling shutter / time offset) → artifacts σε κίνηση.
- 🎬 Auto-exposure mismatch → αστάθεια matching cost (χρησιμοποίησε normalization / Census).
- 🎬 Occlusions στα όρια αντικειμένων → “τρύπες” (holes) που θέλουν ειδική μεταχείριση.
- 🎬 Αξιολόγηση χωρίς σωστό masking (sky, reflective, transparent) → παραπλανητικά metrics.

Rule of thumb: πρώτα “γεωμετρία σωστή”, μετά tuning στον matcher.

- 1) Γιατί το stereo “δουλεύει καλύτερα” σε κοντινές αποστάσεις;
- 2) Τι κερδίζουμε με rectification; Τι προϋποθέτει;
- 3) Σε SfM, γιατί υπάρχει scale ambiguity; Πώς τη λύνεις;
- 4) Δώσε 2 αιτίες outliers στα matches και 1 τρόπο αντιμετώπισης (robust).

- 🎬 Depth estimation = γεωμετρία + αντιστοίχιση: calibration/rectification είναι θεμέλια.
- 🎬 Stereo: disparity \rightarrow depth με $Z=(B \cdot f)/d$. Κοντά = καλύτερα, μακριά = αβεβαιότητα.
- 🎬 SGM/SGBM: πρακτικός συμβιβασμός για dense disparity σε real-time.
- 🎬 3D reconstruction: triangulation + (σε multi-view) SfM + Bundle Adjustment.
- 🎬 Monocular depth: χρήσιμο, αλλά προσοχή στην κλίμακα και στη γενίκευση.

Κύριες πηγές:

Brown.edu Introduction to computer vision cs143: <https://cs.brown.edu/courses/cs143>

OpenCV Depth Map from Stereo Images: https://docs.opencv.org/4.x/dd/d53/tutorial_py_depthmap.html

OpenCV Camera Calibration: https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html

Stanford CS231A Epipolar Geometry (PDF): https://web.stanford.edu/class/cs231a/course_notes/03-epipolar-geometry.pdf

KITTI Stereo benchmark: https://www.cvlibs.net/datasets/kitti/eval_stereo.php

Middlebury Stereo page: <https://vision.middlebury.edu/stereo/>

ETH3D overview: <https://www.eth3d.net/overview>

MiDaS (monocular depth): <https://github.com/isl-org/MiDaS>